

## Trabajo Fin de Máster

Estudio de un rodamiento magnético activo axial  
mediante simulaciones de elementos finitos

Autor/es

Iker Xabier Iturrate Gil

Director/es

Jesús Letosa Fleta

Escuela de Ingeniería y Arquitectura de Zaragoza  
2012

# Estudio de un rodamiento magnético activo axial mediante simulaciones de elementos finitos

## ---RESUMEN---

En este Trabajo Fin de Máster (TFM) se ha realizado el estudio de un rodamiento magnético activo axial mediante simulaciones de elementos finitos.

Así, en primer lugar, se ha realizado una revisión del estado del arte sobre rodamientos magnéticos para estudiar las topologías que se utilizan en el diseño de los mismos.

Tras esto, se ha simulado un modelo mediante el programa *Finite Element Method Magnetic* (FEMM v4.2) a partir de uno de los modelos publicados.

Una vez se dispone del modelo simulado, se han realizado una serie de pruebas de validez para analizar la sensibilidad de los resultados obtenidos frente a diversas modificaciones en el entorno de FEMM. Asimismo, se ha comparado el modelo simulado con el modelo validado mediante varios estudios para evaluar su comportamiento electromagnético.

Después, se ha profundizado en el estudio paramétrico de la geometría que define el rodamiento magnético, con el objetivo de simular un modelo mediante FEMM que mejore las prestaciones del modelo validado. Para ello, se ha implementado un algoritmo en Matlab vR2011 que ejecuta FEMM y permite el diseño y análisis de rodamientos magnéticos activos axiales, basados en la topología estudiada, pero con los parámetros geométricos que se deseen. Así, con esta herramienta diseñada en Matlab, se ha podido llevar a cabo el estudio paramétrico del rodamiento de una manera más visual y rápida. En consecuencia, tras el estudio de diversas modificaciones en la geometría del rodamiento magnético, se ha simulado un rodamiento magnético activo axial que aporta más fuerza axial que el modelo validado.

Por último, se ha simulado un modelo mediante FEMM de un rodamiento magnético activo axial que permite compensar las fuerzas resultantes en sentido axial para un prototipo operativo de rotor levitado.

**I****ÍNDICE GENERAL**

<b>1.</b>	<b>INTRODUCCIÓN.....</b>	<b>5</b>
<b>2.</b>	<b>MODELO DE ELEMENTOS FINITOS DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL .....</b>	<b>6</b>
2.1	ESTRUCTURA Y FUNCIONAMIENTO DEL RODAMIENTO MAGNÉTICO .....	6
2.2	CRITERIOS INICIALES DE DISEÑO CON FEMM.....	10
2.3	PARÁMETROS DEL DISEÑO DEL MODELO MEDIANTE FEMM .....	12
2.4	MATERIALES DEL RODAMIENTO MAGNÉTICO .....	12
<b>3.</b>	<b>PRUEBAS DE VALIDEZ DEL MODELO SIMULADO DEL RODAMIENTO MAGNÉTICO .....</b>	<b>13</b>
3.1	ESTUDIO DE SENSIBILIDAD DE RESULTADOS FRENTE A DISTINTAS CONDICIONES DE MALLADO .....	14
3.2	ESTUDIO DE SENSIBILIDAD DE RESULTADOS FRENTE A MATERIALES DE DISTINTA PERMEABILIDAD .....	15
3.3	ESTUDIO DE SENSIBILIDAD DE RESULTADOS FRENTE A ALARGAMIENTO DE LA FRONTERA.....	16
3.4	ESTUDIO DE FUERZA AXIAL FRENTE AL GAP .....	18
3.5	ESTUDIO DE FUERZA AXIAL FRENTE A LA CORRIENTE.....	22
<b>4.</b>	<b>METODOLOGÍA EMPLEADA EN EL ESTUDIO PARAMÉTRICO DEL RODAMIENTO MAGNÉTICO ....</b>	<b>24</b>
4.1	INTERFAZ GRÁFICA DEL ALGORITMO DE MATLAB .....	24
4.2	CONTENIDO DEL ALGORITMO DE MATLAB.....	27
4.3	RESULTADOS DEL ALGORITMO DE MATLAB.....	27
4.4	EJEMPLOS DE FUNCIONAMIENTO DE LOS ALGORITMOS DE MATLAB.....	28
<b>5.</b>	<b>RESULTADOS DEL ESTUDIO PARAMÉTRICO DEL RODAMIENTO MAGNÉTICO .....</b>	<b>30</b>
5.1	MODIFICACIÓN DE LOS DIÁMETROS .....	30
5.2	MODIFICACIÓN DE ANCHURAS.....	32
<b>6.</b>	<b>DISEÑO DE UN RODAMIENTO MAGNÉTICO ACTIVO AXIAL PARA UN VOLANTE DE INERCIA .....</b>	<b>38</b>
6.1	DESCRIPCIÓN DEL PROTOTIPO OPERATIVO DE ROTOR LEVITADO .....	38
6.2	ESTUDIO A REALIZAR EN EL PROTOTIPO OPERATIVO DE ROTOR LEVITADO .....	43
<b>7.</b>	<b>CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>46</b>
<b>8.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>47</b>
<b>9.</b>	<b>ANEXOS .....</b>	<b>48</b>
9.1	CÓDIGO FUENTE DEL ALGORITMO DE MATLAB “FAXIAL_VS_GAP.M” .....	48
9.2	CÓDIGO FUENTE DEL ALGORITMO DE MATLAB “FAXIAL_VS_CORRIENTE.M” .....	68
9.3	CÓDIGO FUENTE DE LA FUNCIÓN DE MATLAB “COMPROBACIÓN_ERRORES.M” DE “FAXIAL_VS_GAP.M” .....	87
9.4	CÓDIGO FUENTE DE LA FUNCIÓN DE MATLAB “COMPROBACIÓN_ERRORES2.M” DE “FAXIAL_VS_CORRIENTE.M” .....	88

## II

## ÍNDICE DE FIGURAS

<b>FIGURA 2.1.</b> RODAMIENTO MAGNÉTICO ACTIVO AXIAL REALIZADO MEDIANTE AUTOCAD CON (A) TEXTURAS A COLOR; (B) TEXTURAS REALES. ....	7
<b>FIGURA 2.2.</b> SECCIÓN TRANSVERSAL DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL REALIZADA MEDIANTE AUTOCAD CON (A) TEXTURAS A COLOR; (B) TEXTURAS REALES. ....	7
<b>FIGURA 2.3.</b> BUCLE DE CONTROL BÁSICO DE UN RODAMIENTO MAGNÉTICO Y SUS ELEMENTOS. ....	8
<b>FIGURA 2.4.</b> COMPARACIÓN DE FUERZAS: (A) MUELLE MECÁNICO; (B) ELECTROIMÁN (CON UNA CORRIENTE CONSTANTE); (C) ELECTROIMÁN (CON UN GAP CONSTANTE). ....	9
<b>FIGURA 2.5.</b> ALZADO DE LA SECCIÓN TRANSVERSAL DEL RODAMIENTO MAGNÉTICO. ....	11
<b>FIGURA 2.6.</b> PARÁMETROS GEOMÉTRICOS DEL RODAMIENTO MAGNÉTICO. ....	11
<b>FIGURA 3.1.</b> ESTUDIO DE SENSIBILIDAD DE RESULTADOS CON DISTINTAS CONDICIONES DE MALLADO. GRÁFICA DE FUERZA AXIAL FRENTE AL GAP CON UNA CORRIENTE CONSTANTE DE 2 A. ....	14
<b>FIGURA 3.2.</b> ESTUDIO DE SENSIBILIDAD DE RESULTADOS CON MATERIALES DE DISTINTA PERMEABILIDAD. GRÁFICA DE FUERZA AXIAL FRENTE AL GAP CON UNA CORRIENTE CONSTANTE DE 2 A. ....	16
<b>FIGURA 3.3.</b> ENTORNO DE FEMM CON UNA FRONTERA DE RADIO EL DOBLE QUE EL RADIO DEL ESTATOR. ....	17
<b>FIGURA 3.4.</b> ESTUDIO DE SENSIBILIDAD DE RESULTADOS CON FRONTERAS DE DISTINTO TAMAÑO. GRÁFICA DE FUERZA AXIAL FRENTE AL GAP CON UNA CORRIENTE CONSTANTE DE 2 A. ....	18
<b>FIGURA 3.5.</b> SIMULACIÓN EN FEMM DE LA FUERZA AXIAL FRENTE AL GAP CON UNA CORRIENTE DE 1 A Y UN GAP DE 3 MM EN LA QUE SE VISUALIZA LA (A) TRAYECTORIA DE LAS LÍNEAS DE FLUJO MAGNÉTICO EN EL RODAMIENTO MAGNÉTICO; (B) VALORES DEL MÓDULO DE INDUCCIÓN MAGNÉTICA. ....	19
<b>FIGURA 3.6.</b> GRÁFICA DE FUERZA AXIAL FRENTE AL GAP CON DISTINTAS CURVAS DE CORRIENTE OBTENIDA CON FEMM. ....	20
<b>FIGURA 3.7.</b> GRÁFICA DE FUERZA AXIAL FRENTE A LA CORRIENTE CON DISTINTAS CURVAS DE GAP OBTENIDA CON FEMM. ....	23
<b>FIGURA 4.1.</b> INTERFAZ GRÁFICA CREADA MEDIANTE OCTAVE PARA EL ALGORITMO "FAXIAL_vs_CORRIENTE.M". ....	26
<b>FIGURA 4.2.</b> INTERFAZ GRÁFICA CREADA EN MATLAB PARA EL ALGORITMO "FAXIAL_vs_CORRIENTE.M". ....	26
<b>FIGURA 4.3.</b> OPERACIONES REALIZADAS MEDIANTE LA INTERFAZ GRÁFICA DEL ALGORITMO "FAXIAL_vs_CORRIENTE.M" MOSTRANDO: (A) VALIDACIÓN DE ERRORES; (B) MENSAJE DE EJECUCIÓN DE FEMM; (C) CÁLCULO DE LOS PARÁMETROS DEL DEVANADO. ....	26
<b>FIGURA 4.4.</b> INTERFAZ GRÁFICA CON LOS PARÁMETROS DEL EJEMPLO REALIZADO CON EL ALGORITMO "FAXIAL_vs_GAP.M" CON UNA CORRIENTE CONSTANTE DE 2 A. ....	28
<b>FIGURA 4.5.</b> EJECUCIÓN DE UN EJEMPLO PARA EL ALGORITMO "FAXIAL_vs_GAP.M" CON UNA CORRIENTE DE 2 A MOSTRANDO: (A) CÁLCULO DE LOS PARÁMETROS DEL DEVANADO; (B) DISEÑO GRÁFICO OBTENIDO CON FEMM. ....	29
<b>FIGURA 4.6.</b> EJECUCIÓN DE UN EJEMPLO PARA EL ALGORITMO "FAXIAL_vs_GAP.M" CON UNA CORRIENTE DE 2 A MOSTRANDO: (A) GRÁFICA DE LA FUERZA AXIAL FRENTE AL GAP; (B) VALORES DEL GRÁFICO DE FUERZA AXIAL FRENTE AL GAP. ....	29
<b>FIGURA 4.7.</b> EJECUCIÓN DE UN 2º EJEMPLO PARA EL ALGORITMO "FAXIAL_vs_CORRIENTE.M" CON GAP DE 2 MM: (A) GRÁFICA DE LA FUERZA AXIAL FRENTE A LA CORRIENTE; (B) VALORES DEL GRÁFICO DE FUERZA AXIAL FRENTE A LA CORRIENTE. ....	29
<b>FIGURA 5.1.</b> FUERZA AXIAL FRENTE A LA CORRIENTE CON UN GAP DE 3 MM PARA RODAMIENTOS CON MODIFICACIONES EN LOS DIÁMETROS. ....	31

<b>FIGURA 5.2.</b> FUERZA AXIAL FRENTE AL GAP CON UN CORRIENTE DC DE 3 A PARA RODAMIENTOS CON MODIFICACIONES EN LOS DIÁMETROS. ....	31
<b>FIGURA 5.3.</b> DISEÑO GRÁFICO DEL RODAMIENTO MAGNÉTICO OBTENIDO EN FEMM CON EL ALGORITMO "FAXIAL_VS_CORRIENTE.M" (A) SIN CUERNOS EN EL RODAMIENTO; (B) SIN CUERNO DEL DISCO ROTÓRICO Y (C) SIN CUERNO DEL DISCO DEL ESTATOR. ....	32
<b>FIGURA 5.4.</b> FUERZA AXIAL FRENTE A LA CORRIENTE CON UN GAP DE 3 MM PARA RODAMIENTOS CON MODIFICACIONES EN LAS ANCHURAS. ....	33
<b>FIGURA 5.5.</b> FUERZA AXIAL FRENTE AL GAP CON UN CORRIENTE DC DE 3 A PARA RODAMIENTOS CON MODIFICACIONES EN LAS ANCHURAS. ....	33
<b>FIGURA 5.6.</b> DISEÑO GRÁFICO DEL RODAMIENTO MAGNÉTICO OBTENIDO EN FEMM CON EL ALGORITMO "FAXIAL_VS_CORRIENTE.M" (A) CON ANILLOS MÁS GRUESOS, (B) CON ANILLOS MÁS ESTRECHOS Y (C) CON ANILLO INTERNO MÁS ESTRECHO Y ANILLO EXTERNO MÁS ANCHO. ....	34
<b>FIGURA 6.1.</b> ESTRUCTURA DEL PROTOTIPO OPERATIVO DE ROTOR LEVITADO DE [8]. ....	38
<b>FIGURA 6.2.</b> SECCIÓN TRANSVERSAL DE LA ESTRUCTURA DEL ROTOR DEL PROTOTIPO OPERATIVO DE ROTOR LEVITADO REALIZADA CON AUTOCAD. ....	39
<b>FIGURA 6.3.</b> ACOPLAMIENTO DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL AL PROTOTIPO OPERATIVO DE ROTOR LEVITADO. ....	40
<b>FIGURA 6.4.</b> CARACTERÍSTICA DE FUERZA AXIAL FRENTE AL DESPLAZAMIENTO AXIAL DEL PROTOTIPO DE ROTOR LEVITADO EN AUSENCIA DEL PESO DEL ROTOR. ....	41
<b>FIGURA 6.5.</b> GRÁFICA DEL PESO DEL ROTOR DEL PROTOTIPO DE ROTOR LEVITADO FRENTE AL DESPLAZAMIENTO AXIAL. ....	41
<b>FIGURA 6.6.</b> CARACTERÍSTICA DE FUERZA AXIAL FRENTE AL DESPLAZAMIENTO AXIAL DEL PROTOTIPO DE ROTOR LEVITADO EN CONSIDERANDO EL PESO DEL ROTOR. ....	42
<b>FIGURA 6.7.</b> GRÁFICA DE FUERZA AXIAL FRENTE AL GAP CON UNA CORRIENTE CONSTANTE DE 3 A (IZQUIERDA) Y FRENTE A LA CORRIENTE CON EL GAP DEL PUNTO DE EQUILIBRIO DE 1 MM (DERECHA) OBTENIDA CON FEMM EN UN RODAMIENTO MAGNÉTICO ACTIVO AXIAL DISEÑADO PARA EL PROTOTIPO DE [8]. ....	44
<b>FIGURA 6.8.</b> CÁLCULO DEL DEVANADO DEL MODELO SIMULADO DE RODAMIENTO MAGNÉTICO ACTIVO AXIAL PARA EL PROTOTIPO DE [8]. ....	45

## III

## ÍNDICE DE TABLAS

<b>TABLA 2.1.</b> PARÁMETROS DEL MODELO VALIDADO DEL RODAMIENTO MAGNÉTICO [1]. .....	12
<b>TABLA 3.1.</b> TAMAÑO DE LAS FRONTERAS ANALIZADAS PARA EL ESTUDIO DE SENSIBILIDAD DE RESULTADOS. ....	16
<b>TABLA 5.1.</b> PARÁMETROS DEL MODELO DEL RODAMIENTO MAGNÉTICO CON ANILLOS ELECTROMAGNÉTICOS DE 10 MM DE ANCHURA. ....	35
<b>TABLA 5.2.</b> MASA DEL DISCO ROTÓRICO SEGÚN EL MODELO PUBLICADO EN [1] Y SEGÚN EL MODELO SIMULADO CON ANILLOS ELECTROMAGNÉTICOS DE 10 MM. ....	36
<b>TABLA 5.3.</b> COMPARACIÓN DE FUERZA AXIAL OBTENIDA FRENTE AL GAP ENTRE EL MODELO PUBLICADO EN [1] Y EL MODELO SIMULADO CON ANILLOS DE 10 MM CON UNA CORRIENTE CONSTANTE DC DE 3 A. ....	36
<b>TABLA 5.4.</b> COMPARACIÓN DE FUERZA AXIAL OBTENIDA FRENTE A LA CORRIENTE ENTRE EL MODELO PUBLICADO EN [1] Y EL MODELO SIMULADO CON ANILLOS DE 10 MM CON UN GAP CONSTANTE DE 3 MM. ....	37
<b>TABLA 6.1.</b> PARÁMETROS DEL ROTOR DEL PROTOTIPO DE [8].....	43
<b>TABLA 6.2.</b> PARÁMETROS DEL MODELO DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL PARA EL PROTOTIPO OPERATIVO DE ROTOR LEVITADO. ....	43

## 1. INTRODUCCIÓN

Los sistemas de rodamientos magnéticos son indispensables para el funcionamiento de máquinas rotativas de alta velocidad. Presentan varias ventajas respecto a los rodamientos clásicos de bolas, como son el funcionamiento sin rozamiento, ausencia de mantenimiento o baja emisión de ruido. Asimismo, su potencial para proporcionar altas velocidades sin rozamiento hace de este tipo de rodamientos el candidato ideal para algunas aplicaciones industriales como: bombas de vacío, discos duros, centrifugadoras de alta velocidad, volantes de inercia, etc.

Actualmente, se han diseñado muchas configuraciones para estabilizar el movimiento del rotor en sus distintos grados de libertad, que genéricamente pueden dividirse en sistemas activos [1][2] y pasivos [3][4], según utilicen o no mecanismos de control de la fuerza ejercida. Según el grado de libertad del rotor sobre el que actúen, hablaremos de rodamientos magnéticos radiales [5][6] o axiales [1][7].

El objetivo principal de este TFM es desarrollar un modelo electromagnético para rodamientos magnéticos activos de fuerza axial de la topología descrita en [1]. Dicha topología a estudiar, se ha elegido por ser de interés en el proyecto de investigación [8], tras un estudio de la bibliografía sobre el tema. Con el modelo desarrollado, se quieren estudiar las prestaciones de este tipo de rodamiento en función de los parámetros geométricos y materiales que lo definen. Así, los problemas que se abordan en este TFM y los objetivos que se desean cumplir se citan a continuación:

- Revisión del estado del arte sobre rodamientos magnéticos activos axiales.
- Diseño e implementación de un modelo electromagnético del rodamiento magnético activo axial mediante un programa de elementos finitos en 2D que permita analizar su funcionamiento.
- Comparación del modelo desarrollado con un modelo validado.
- Estudio paramétrico del comportamiento de la topología analizada para el rodamiento magnético.

En primer lugar, se desarrolla un modelo electromagnético de un rodamiento magnético activo axial basado en el modelo validado descrito en [1] mediante el programa de elementos finitos *Finite Element Method Magnetic* (FEMM v4.2).

Tras esto, se compara el modelo simulado con el modelo validado, descrito en [1], para analizar el comportamiento del rodamiento.

Una vez verificada la aproximación del modelo simulado con los datos publicados [1], se ha implementado un algoritmo mediante Matlab que permite el diseño de rodamientos magnéticos activos axiales basados en la topología estudiada con los parámetros geométricos que se deseen. Este algoritmo permite realizar un estudio paramétrico del rodamiento magnético de una manera más rápida y dinámica.

Después, en la sección de resultados y mediante el algoritmo de Matlab, se ha conseguido simular un modelo de un rodamiento magnético activo axial que proporciona más fuerza axial que el rodamiento descrito en el modelo validado.

Por último, se describe el estudio realizado para un prototipo operativo de rotor levitado desarrollado en [8] que estaba diseñado para compensar el movimiento del rotor en sentido radial mediante imanes permanentes. Así, se simula un modelo de un rodamiento magnético activo axial que permite compensar el peso del rotor de [8] gracias a la fuerza axial que ejerce el rodamiento del nuevo modelo simulado.

## **2. MODELO DE ELEMENTOS FINITOS DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL**

Se va a desarrollar un modelo de un rodamiento magnético activo axial en 2D mediante FEMM a partir del modelo validado descrito en [1] para analizar su comportamiento electromagnético.

### **2.1 Estructura y funcionamiento del rodamiento magnético**

#### **2.1.1 Estructura del rodamiento magnético**

Se pretende modelar un rodamiento magnético activo axial cuya **estructura** está compuesta por un estator, un devanado estatístico, un rotor y un entrehierro. Estas partes constituyen el circuito magnético electromagnético del modelo, cuyas características se citan a continuación:

- Estator: Es una carcasa cilíndrica de material ferromagnético, como el acero, compuesta por dos anillos electromagnéticos.
- Devanado estatístico: Es una bobina de cobre que se aloja entre los anillos electromagnéticos del estator y es la encargada de generar el campo magnético mediante una corriente continua (DC) que circula a su través.
- Rotor: Es un disco de material ferromagnético, como el acero, que no permanece estático, sino que gira respecto de su eje. Se encuentra separado axialmente del

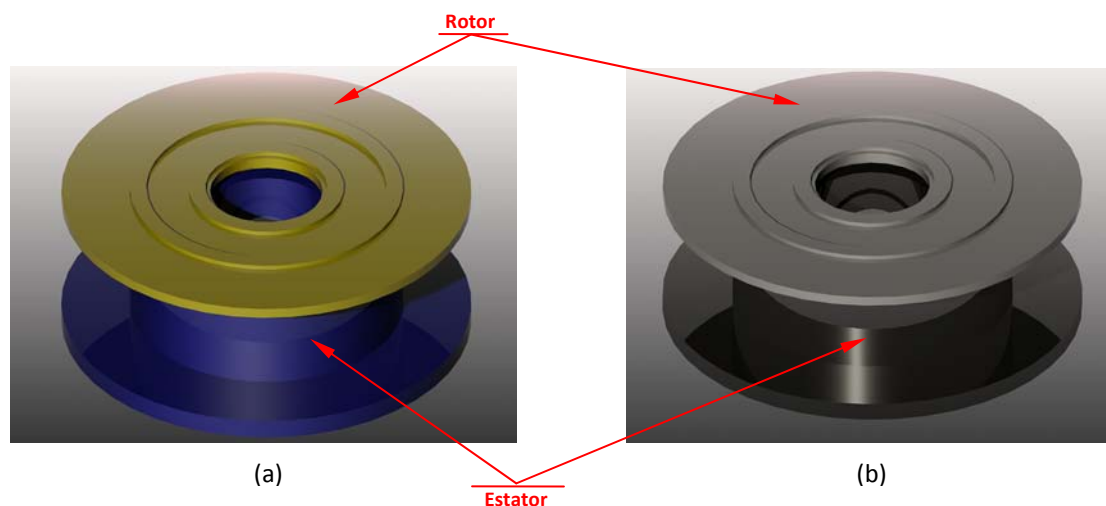


estator por una distancia denominada gap, la cual actúa como entrehierro del circuito magnético.

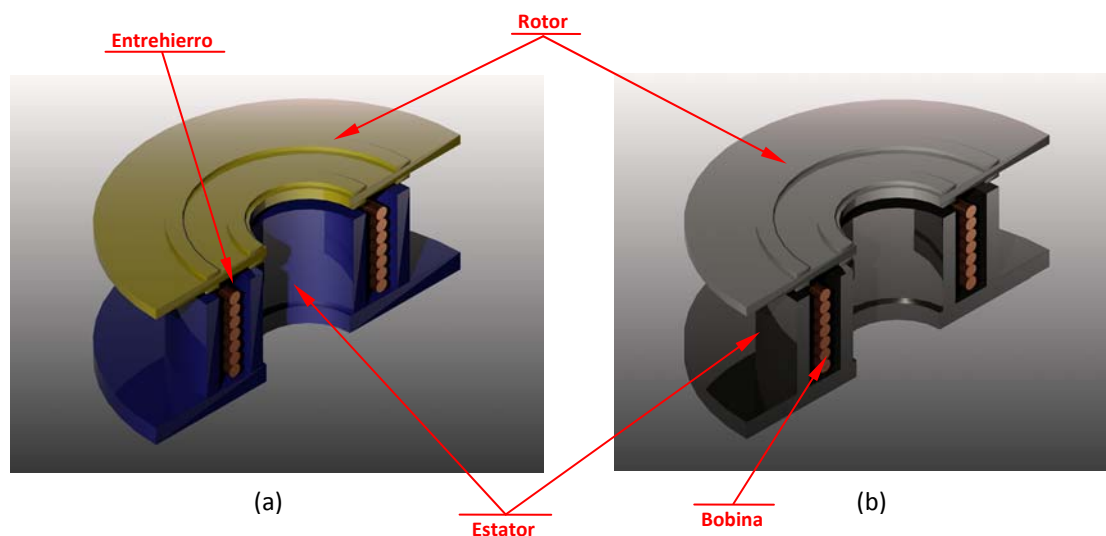
- Entrehierro: Es el medio material no magnético que separa al disco rotórico del estator, cuya distancia la marca el gap. En este caso, el medio material es el aire.

En la Figura 2.1 se ilustra un modelo en 3D del rodamiento magnético activo axial realizado con el programa de diseño gráfico Autocad v2012. En dichas figuras, se refleja el rodamiento con texturas a color cuyos contrastes permiten distinguir mejor las partes del rodamiento (Figura 2.1a) y una vista del rodamiento con texturas reales, con materiales derivados del acero (Figura 2.1b).

La bobina y el entrehierro se pueden visualizar en Autocad al aplicar una sección transversal a los rodamientos ilustrados anteriormente (Figura 2.1), tal y como se refleja en la Figura 2.2.



**Figura 2.1.** Rodamiento magnético activo axial realizado mediante Autocad con (a) texturas a color; (b) texturas reales.



**Figura 2.2.** Sección transversal del rodamiento magnético activo axial realizada mediante Autocad con (a) texturas a color; (b) texturas reales.

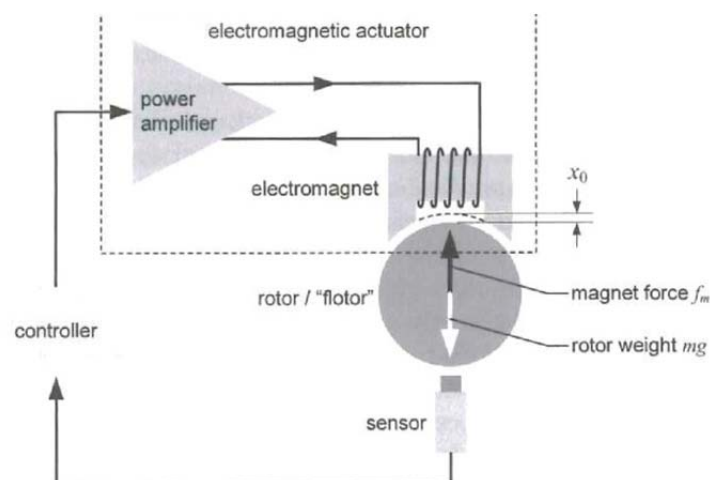
### 2.1.2 Funcionamiento del rodamiento magnético

Conocida la estructura del rodamiento, se debe explicar su **funcionamiento**. Éste se basa en el principio de los rodamientos magnéticos activos, que son sistemas que sustentan ejes rotativos sin contacto físico al mantener el rotor en el aire mediante una fuerza magnética controlada eléctricamente [9].

El término “activo” implica que la fuerza magnética es controlada activamente por electroimanes, por un bucle de control de realimentación adecuado y otros elementos como sensores y amplificadores de potencia [10].

En la Figura 2.3 se ilustra un sencillo ejemplo de un bucle de control de un rodamiento magnético activo. En ella se refleja un cuerpo rígido en suspensión (rotor) con un solo grado de libertad y corresponde a un rodamiento magnético real muy simplificado (puesto que no se tienen en cuenta ni los efectos rotativos ni el movimiento transversal del rotor). No obstante, este ejemplo permite entender las propiedades básicas del bucle de control de un rodamiento magnético. A continuación, se describen brevemente los elementos de este sistema de control [10]:

- Rotor: es la masa que se pretende sustentar a una distancia  $x_0$  del electroimán.
- Sensor de posición: mide la desviación entre la posición deseada del rotor  $x_0$  y la posición actual  $x$  y transmite dicha información al controlador.
- Controlador: se encarga de mantener la posición del rotor en el valor deseado. Envía una señal de posición del rotor al amplificador de potencia.
- Amplificador de potencia: recibe la señal del controlador y la transforma en una consigna de corriente eléctrica para que circule por la bobina. Esa bobina, mediante el electroimán, será la encargada de generar la fuerza magnética.



**Figura 2.3.** Bucle de control básico de un rodamiento magnético y sus elementos.

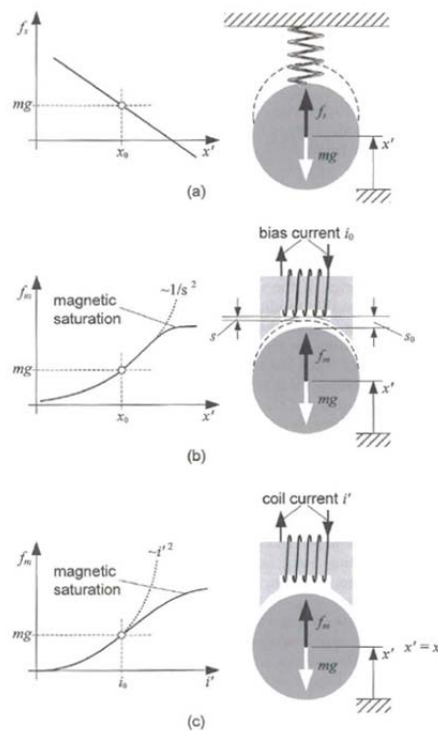
Para entender mejor el funcionamiento de un modelo simple de rodamiento magnético, se puede comparar con el sistema mecánico de un muelle [10], tal y como se refleja en la Figura 2.4. De aquí se deducen las siguientes ideas:

- La fuerza mecánica del muelle  $f_s$  y la fuerza magnética del rodamiento  $f_m$  se comportan de un modo distinto: por un lado, se aprecia que la fuerza mecánica del muelle  $f_s$  disminuye conforme aumenta la distancia con el suelo  $x'$  y produce un efecto de repulsión que estabiliza el movimiento del rotor en torno al punto de equilibrio  $(x_0, s_0)$ ; mientras, la fuerza magnética  $f_m$  aumenta drásticamente conforme disminuye el gap  $s$ , lo que da lugar a un comportamiento inestable.
- La rigidez mecánica  $k$  de la suspensión es la derivada de la fuerza de suspensión respecto del desplazamiento, tal y como se refleja en la siguiente expresión (1).

$$k = -df/dx \quad (1)$$

Matemáticamente, el signo de la rigidez mecánica en el punto de operación  $(x_0, i_0, mg)$  determina la estabilidad de esta posición de equilibrio.

- La fuerza magnética es inversamente proporcional al cuadrado del gap y proporcional al cuadrado de la corriente que circula por la bobina. Para pequeños gaps o corrientes elevadas, respectivamente, el flujo magnético puede llegar saturarse, dando lugar a un comportamiento no lineal de la fuerza magnética.



**Figura 2.4.** Comparación de fuerzas: (a) muelle mecánico; (b) electroimán (con una corriente constante); (c) electroimán (con un gap constante).

Así, el control debe aportar una fuerza de restauración, que sea similar, por ejemplo, a la fuerza mecánica de un muelle. Además, la fuerza de control debe proporcionar una componente de amortiguamiento para atenuar las oscilaciones en torno al punto de operación. Así, una aproximación que tiene en cuenta este criterio para obtener la fuerza de control deseada da lugar a la siguiente expresión [10].

$$f = -kx - d\dot{x} \quad (2)$$

Siendo:

$f$ : fuerza del rodamiento.

$k$ : rigidez del rodamiento.

$d$ : Amortiguamiento.

$\dot{x}$ : Velocidad de variación del gap (o derivada del gap respecto del tiempo).

Por tanto, con esta explicación del funcionamiento del rodamiento magnético se pretende dar a conocer que la sustentación del rotor en el aire depende de varios factores: compensación de fuerzas, rigidez y amortiguamiento [10]. En este TFM, se ha diseñado un modelo que pretende cubrir uno de estos factores como es la compensación de fuerzas del rodamiento en sentido axial, quedando pendiente para futuros trabajos el estudio de los otros dos factores.

## **2.2 Criterios iniciales de diseño con FEMM**

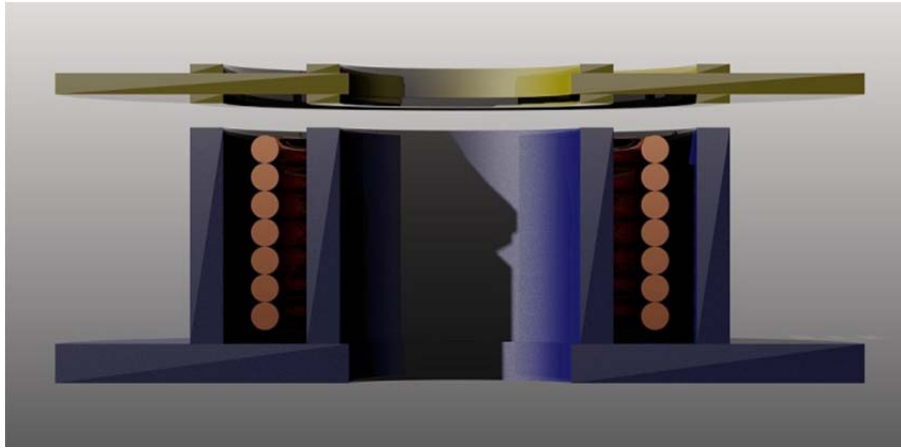
Antes de realizar el modelo con FEMM, se deben tener en cuenta una serie de criterios de diseño que permitan implementar el modelo del rodamiento magnético correctamente.

En primer lugar, se trata de un problema de tipo magnetostático, puesto que se pretende analizar el comportamiento del flujo magnético en un electroimán generado por una bobina recorrida por una corriente eléctrica.

Además, el sistema a representar es un modelo con simetría axisimétrica o axial, puesto que el rodamiento presenta una simetría de revolución o cilíndrica.

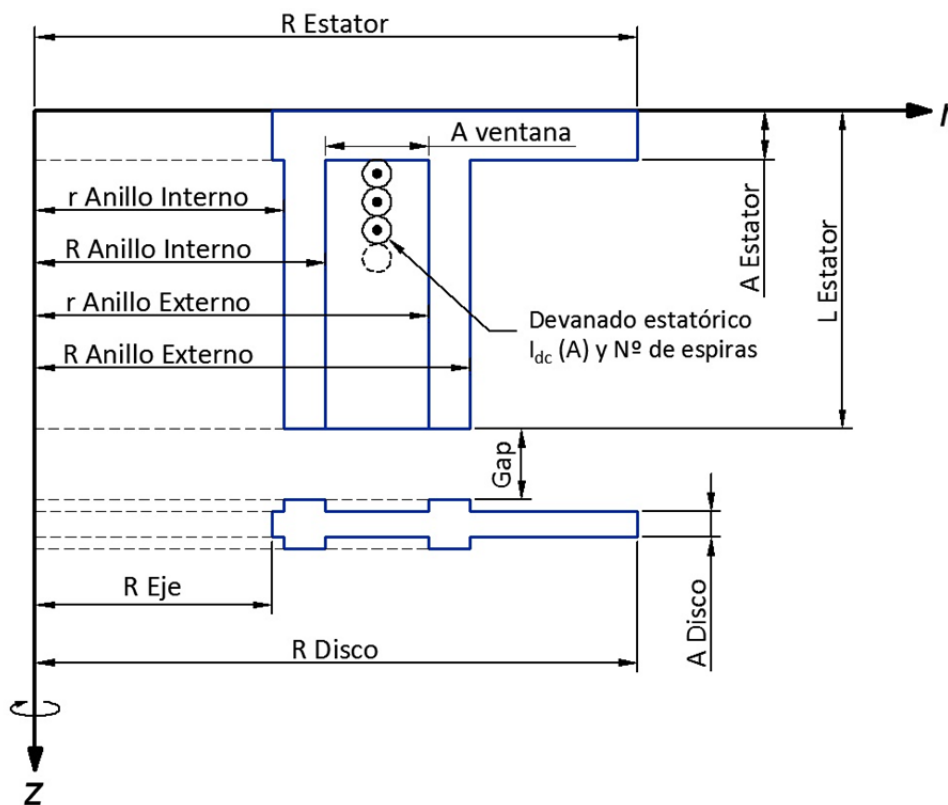
Asimismo, la frecuencia de operación es nula, puesto que se trabaja con corriente continua.

Finalmente, para representar el modelo en FEMM, se dibuja la sección transversal de medio rodamiento, siguiendo el modelo propuesto por [1]. En la Figura 2.5 se representa el alzado de dicha sección, realizada a partir de la vista mostrada en la Figura 2.2 con Autocad.



**Figura 2.5.** Alzado de la sección transversal del rodamiento magnético.

A partir de dicha sección, se pueden visualizar los parámetros geométricos del rodamiento magnético que definen las dimensiones del mismo, tal y como se ilustra en la Figura 2.6. En dicha figura, se aprecia que el Eje Z es el que representa la simetría axial o de revolución del rodamiento.



**Figura 2.6.** Parámetros geométricos del rodamiento magnético.

### 2.3 Parámetros del diseño del modelo mediante FEMM

Una vez fijados los criterios iniciales de diseño, se dibuja el rodamiento magnético en FEMM con las dimensiones del modelo validado propuesto en [1]. Los parámetros de dicho modelo se reflejan en la Tabla 2.1.

**TABLA 2.1.** PARÁMETROS DEL MODELO VALIDADO DEL RODAMIENTO MAGNÉTICO [1].

PARÁMETRO	VALOR
Número de espiras del devanado	200 vueltas
Diámetro de espira	1 mm
Radio del Eje	18,4 mm
Radio del Estator	60,7 mm
Anchura del Estator	6,5 mm
Longitud del Estator	42,3 mm
Radio Interior del Anillo Interno	19,9 mm
Radio Exterior del Anillo Interno	25,4 mm
Radio Interior del Anillo Externo	39,2 mm
Radio Exterior del Anillo Externo	44,7 mm
Radio del Disco Rotórico	60,7 mm
Anchura del Disco Rotórico	4,5 mm

### 2.4 Materiales del rodamiento magnético

El modelo validado propuesto en [1] no especifica los materiales que se utilizan para el diseño del rodamiento magnético.

En primer lugar, para el devanado estático se ha escogido una bobina de cobre.

En segundo lugar, revisando el estado del arte, la mayoría de los rodamientos magnéticos están hechos bien de acero o bien de hierro [11]. Estos materiales presentan valores de permeabilidad relativa superiores a 1000 [11].

Considerando el circuito magnético equivalente del rodamiento, la reluctancia del material se puede traducir como la “dificultad que ofrece dicho material al paso del flujo magnético”. Así, cuanto mayor es la permeabilidad  $\mu$  del material, menor es la reluctancia  $\mathcal{R}$  y, por tanto, más fácilmente circula el flujo magnético, tal y como se aprecia en la expresión (3).

$$\mathcal{R} = \frac{l}{\mu \cdot S} \Rightarrow \mathcal{R} \propto \frac{k_1}{\mu} \quad (3)$$

Siendo:

$\mathcal{R}$ : Reluctancia magnética, en Av/Wb.

$l$ : Longitud de la trayectoria del circuito magnético, en m.

$\mu$ : Permeabilidad del material, adimensional.

$S$ : Sección del material atravesada por el flujo magnético, en m<sup>2</sup>.

$k_1$ : Constante.

Por este motivo, este tipo de materiales son más apropiados para el diseño de rodamientos magnéticos. No obstante, en lugar de escoger un material magnético de la librería de FEMM al azar, se va a realizar un estudio en el siguiente apartado para analizar la sensibilidad de los resultados obtenidos con FEMM para rodamientos magnéticos compuestos por materiales de distinta permeabilidad.

### **3. PRUEBAS DE VALIDEZ DEL MODELO SIMULADO DEL RODAMIENTO MAGNÉTICO**

Para evaluar el comportamiento del modelo simulado de elementos finitos realizado con FEMM, se han realizado una serie de pruebas o casos.

Los tres primeros casos (3.1, 3.2 y 3.3) corresponden a un estudio de sensibilidad de los resultados obtenidos con el modelo simulado en función de una serie de modificaciones en el diseño del modelo. Los dos últimos casos (3.4 y 3.5) corresponden a un estudio comparativo entre los resultados obtenidos con el modelo simulado y las pruebas realizadas según el modelo publicado descrito en [1].

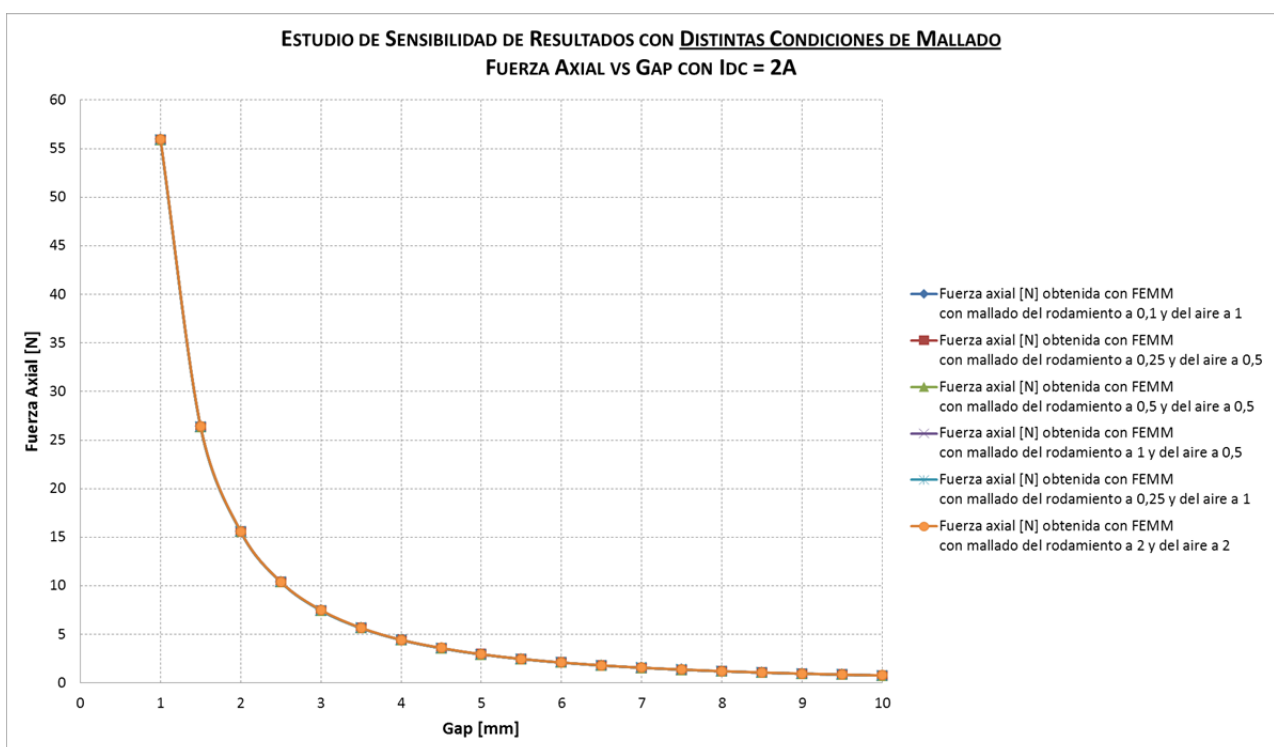
- **Caso 3.1:** Estudio de sensibilidad de resultados del modelo simulado frente a distintas condiciones de mallado en FEMM.
- **Caso 3.2:** Estudio de sensibilidad de los resultados obtenidos con el modelo simulado frente a materiales de distinta permeabilidad.
- **Caso 3.3:** Estudio de sensibilidad de resultados del modelo simulado frente a distintos tamaños de frontera en FEMM.
- **Caso 3.4 [1]:** Estudio de la fuerza axial ejercida por el estator en el rotor, manteniendo una corriente DC constante y variando la distancia de separación o gap entre 1 y 10 mm.
- **Caso 3.5 [1]:** Estudio de la fuerza axial ejercida por el estator en el rotor, manteniendo constante la distancia de separación o gap y variando la corriente DC entre 0 y 3 A.

### 3.1 Estudio de sensibilidad de resultados frente a distintas condiciones de mallado

En este primer caso se va a evaluar la sensibilidad de los resultados obtenidos en un modelo simulado de rodamiento magnético con los parámetros geométricos descritos en la Tabla 2.1 frente a distintas condiciones de mallado en FEMM.

Así, para este caso se va a calcular la fuerza axial ejercida por el estator en el disco rotórico variando la distancia que los separa y manteniendo una corriente constante de 2 A. En la Figura 3.1 se ilustran las distintas curvas de fuerza axial frente al gap obtenidas con FEMM en función del mallado del modelo.

En dicha figura se puede apreciar que los resultados de fuerza axial frente al gap obtenidos con FEMM para las distintas condiciones de mallado son prácticamente los mismos. Por tanto, puesto que cuanto más denso es el mallado más lenta es la ejecución de FEMM, se escoge un mallado de valor 1 para el aire. Para el mallado del rodamiento, se escoge un mallado de valor 0,25, que es inmediatamente superior al mallado más fino analizado.



**Figura 3.1.** Estudio de sensibilidad de resultados con distintas condiciones de mallado. Gráfica de fuerza axial frente al gap con una corriente constante de 2 A.



### **3.2 Estudio de sensibilidad de resultados frente a materiales de distinta permeabilidad**

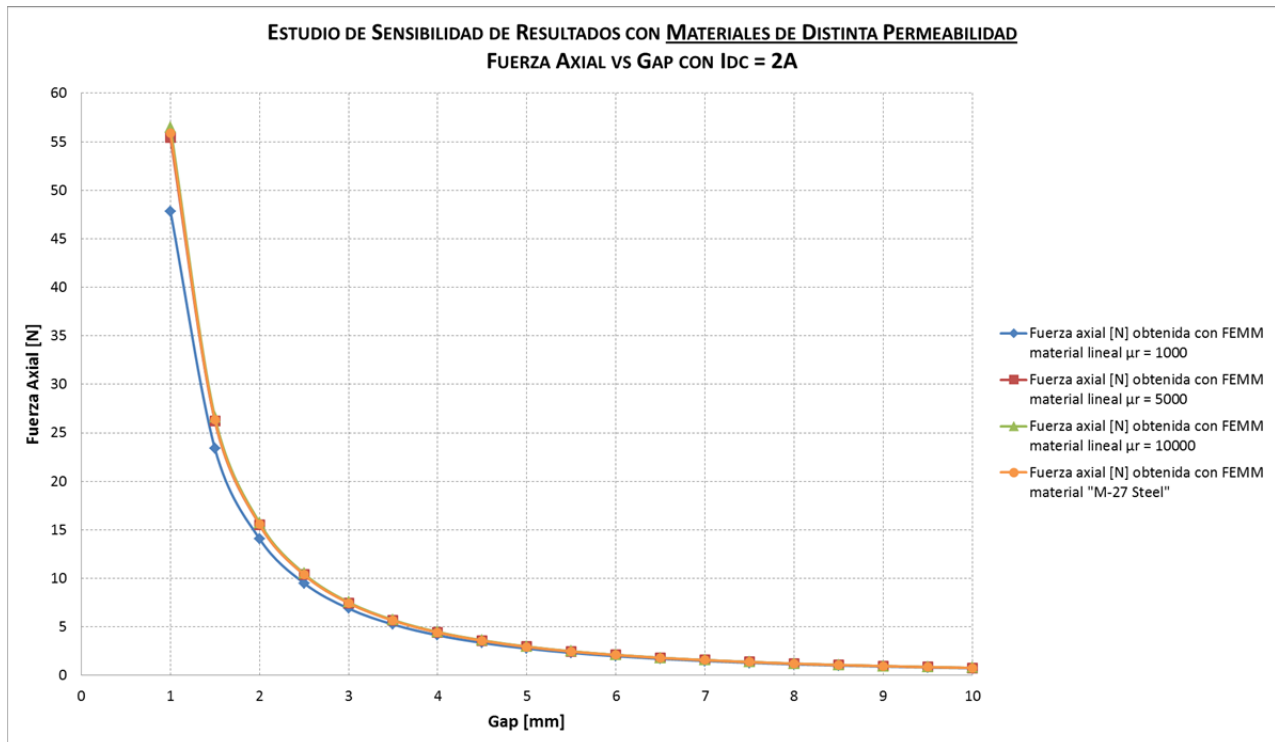
A continuación, se va a evaluar la sensibilidad de los resultados obtenidos en un modelo simulado de rodamiento magnético con los parámetros geométricos descritos en la Tabla 2.1 frente a distintos materiales magnéticos.

Para ello, se van a crear en FEMM varios materiales de distinta permeabilidad relativa (1000, 5000 y 10000) y con característica magnética lineal. Además, se utilizará un material magnético derivado del acero escogido de la librería de FEMM, que en este caso es el material *M-27 Steel* cuya permeabilidad relativa es de 12138.

Así, para este caso se va a calcular la fuerza axial ejercida por el estator en el disco rotórico variando la distancia que los separa y manteniendo una corriente constante de 2 A. En la Figura 3.2 se ilustran las distintas curvas de fuerza axial frente al gap obtenidas con FEMM en función de los materiales del rodamiento magnético.

En dicha Figura 3.2 se puede apreciar que los resultados obtenidos con los rodamientos de permeabilidad relativa 5000, 10000 y el del acero *M-27 Steel* son muy similares. No obstante, se aprecia que para materiales de baja permeabilidad relativa, como el material cuyo valor es de 1000, los resultados varían respecto de los anteriores. Así, se deduce que para materiales poco magnéticos o con baja permeabilidad relativa, se obtienen resultados que varían ligeramente respecto de otros materiales más magnéticos.

Por tanto, a partir de aquí, se escoge el material *M-27 Steel* de la librería de FEMM para el estator y el rotor del rodamiento, por ser un material de elevada permeabilidad magnética cuyo comportamiento es similar a otros materiales con altas permeabilidades comprendidas entre 5000-10000.



**Figura 3.2.** Estudio de sensibilidad de resultados con materiales de distinta permeabilidad.  
Gráfica de fuerza axial frente al gap con una corriente constante de 2 A.

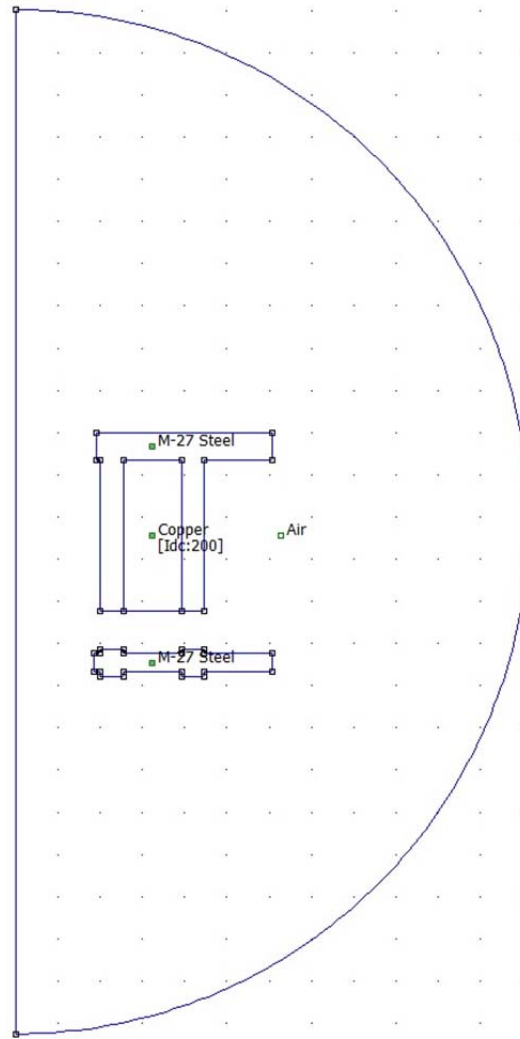
### 3.3 Estudio de sensibilidad de resultados frente a alargamiento de la frontera

En este caso se va a evaluar la sensibilidad de los resultados obtenidos en un modelo simulado de rodamiento magnético con los parámetros geométricos descritos en la Tabla 2.1 frente a distintos tamaños de frontera.

En primer lugar, se van a crear fronteras de varios tamaños donde se situará el rodamiento magnético. Así, las distintas fronteras analizadas están constituidas por un semicírculo cuyos radios se citan en la Tabla 3.1. En la Figura 3.3 se ilustra la frontera 2 en el entorno de FEMM.

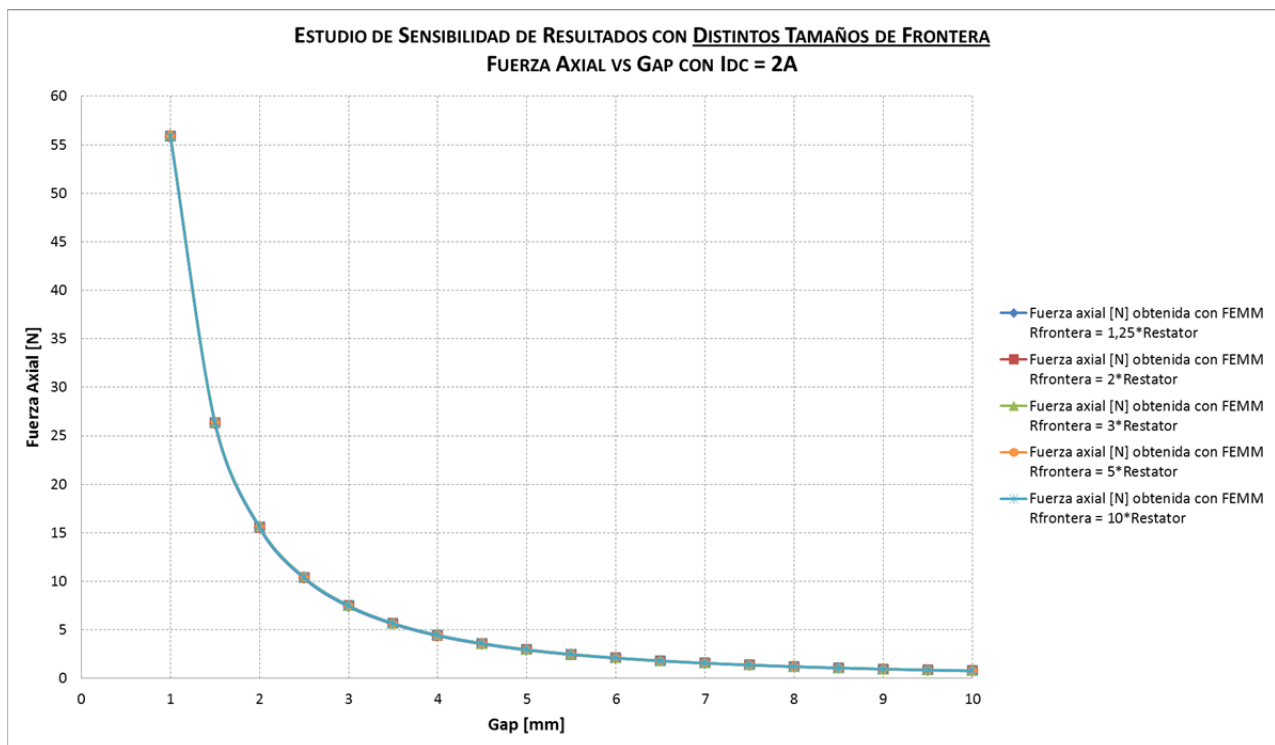
**TABLA 3.1.** TAMAÑO DE LAS FRONTERAS ANALIZADAS PARA EL ESTUDIO DE SENSIBILIDAD DE RESULTADOS.

DENOMINACIÓN DE LA FRONTERA	RADIO DE LA FRONTERA [mm]
<b>Frontera 1</b>	$R_{frontera} = 1,25 \times R_{Estator}$
<b>Frontera 2</b>	$R_{frontera} = 2 \times R_{Estator}$
<b>Frontera 3</b>	$R_{frontera} = 3 \times R_{Estator}$
<b>Frontera 4</b>	$R_{frontera} = 5 \times R_{Estator}$
<b>Frontera 5</b>	$R_{frontera} = 10 \times R_{Estator}$



**Figura 3.3.** Entorno de FEMM con una frontera de radio el doble que el radio del estator.

En la Figura 3.4 se ilustran las distintas curvas de fuerza axial frente al gap con una corriente constante de 2 A obtenidas con FEMM, en función del tamaño de la frontera. Se puede apreciar que los resultados obtenidos son muy similares independientemente del tamaño de la frontera. Así, la mayor zona de concentración de flujo magnético se localiza en el rodamiento magnético y, por tanto, una frontera que abarque dos veces el radio del estator es suficiente. Además, este tamaño de frontera escogido permite una ejecución de FEMM más rápida, puesto que es uno de los tamaños de frontera más pequeño de entre los analizados.



**Figura 3.4.** Estudio de sensibilidad de resultados con fronteras de distinto tamaño.  
Gráfica de fuerza axial frente al gap con una corriente constante de 2 A.

### 3.4 Estudio de fuerza axial frente al gap

Tal y como se ha descrito anteriormente, en este caso se va a calcular la fuerza axial ejercida por el estator en el disco rotórico variando la distancia que los separa.

En el modelo publicado en [1] no se especifica el valor constante de la corriente en DC que se utiliza para obtener los resultados. Por esto, es necesario realizar varias pruebas para averiguar el valor de corriente que se utiliza en el mismo.

El gap inicial será de 1 mm y se irá aumentando, con incrementos de 0,5 mm, hasta llegar a un gap de 10 mm. Para ello, se irá desplazando axialmente el disco del rotor manteniendo fijo el estator.

El procedimiento a seguir en FEMM es el siguiente:

#### 1. Diseño del rodamiento.

Se dibuja el rodamiento magnético en FEMM según los parámetros geométricos descritos en la Tabla 2.1 con los materiales y condiciones de contorno y frontera descritos anteriormente.

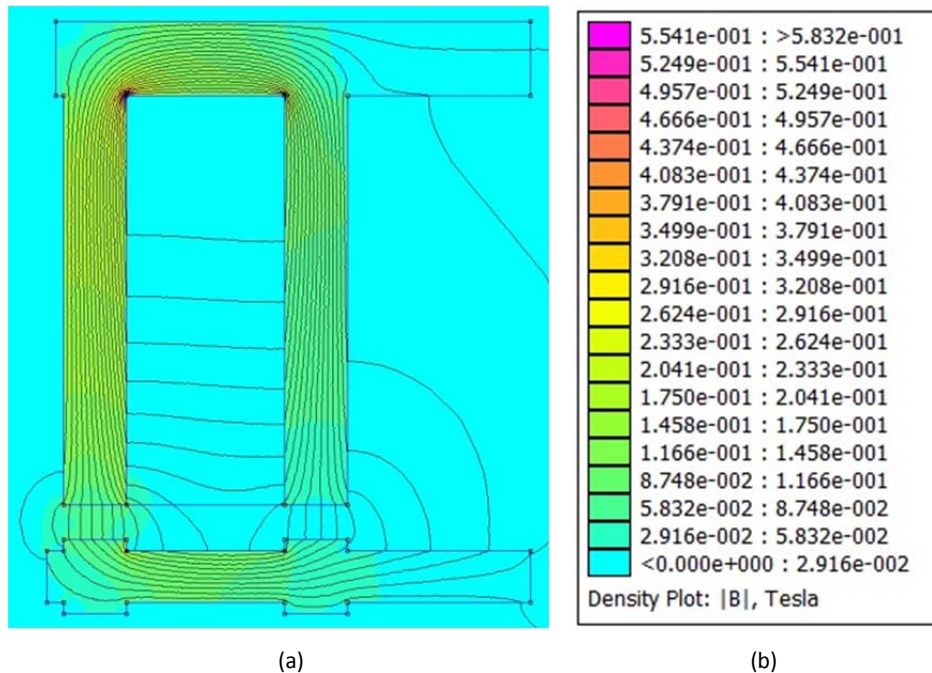
#### 2. Mallado del modelo.

Tras completar el diseño del rodamiento magnético, se realiza el mallado del modelo con FEMM, siguiendo las condiciones descritas en apartados anteriores.

### 3. Inicio del post-procesador.

Tras obtener el mallado, ya se puede calcular la circulación de flujo magnético en el sistema mediante el post-procesador de FEMM.

En la Figura 3.5 se ilustra la trayectoria de las líneas de flujo magnético y los valores de inducción magnética obtenidos con FEMM al calcular la fuerza axial frente al gap con una corriente de 1 A y un gap de 3 mm.



**Figura 3.5.** Simulación en FEMM de la fuerza axial frente al gap con una corriente de 1 A y un gap de 3 mm en la que se visualiza la (a) trayectoria de las líneas de flujo magnético en el rodamiento magnético; (b) valores del módulo de inducción magnética.

### 4. Cálculo de la fuerza axial ejercida por el estator en el rotor frente al gap.

Una vez iniciado el post-procesador de FEMM, se selecciona el bloque del disco rotórico y se calcula la fuerza axial mediante una integral de fuerza.

### 5. Aumento del gap.

Tras disponer del primer valor de fuerza, se incrementa en 0,5 mm el gap separando el disco rotórico axialmente respecto del estator para hacer una nueva medición de fuerza.

### 6. Repetición de los pasos 1-5.

Se repite la toma de medida de la fuerza axial para cada una de las nuevas distancias de gap, hasta alcanzar un gap de 10 mm.

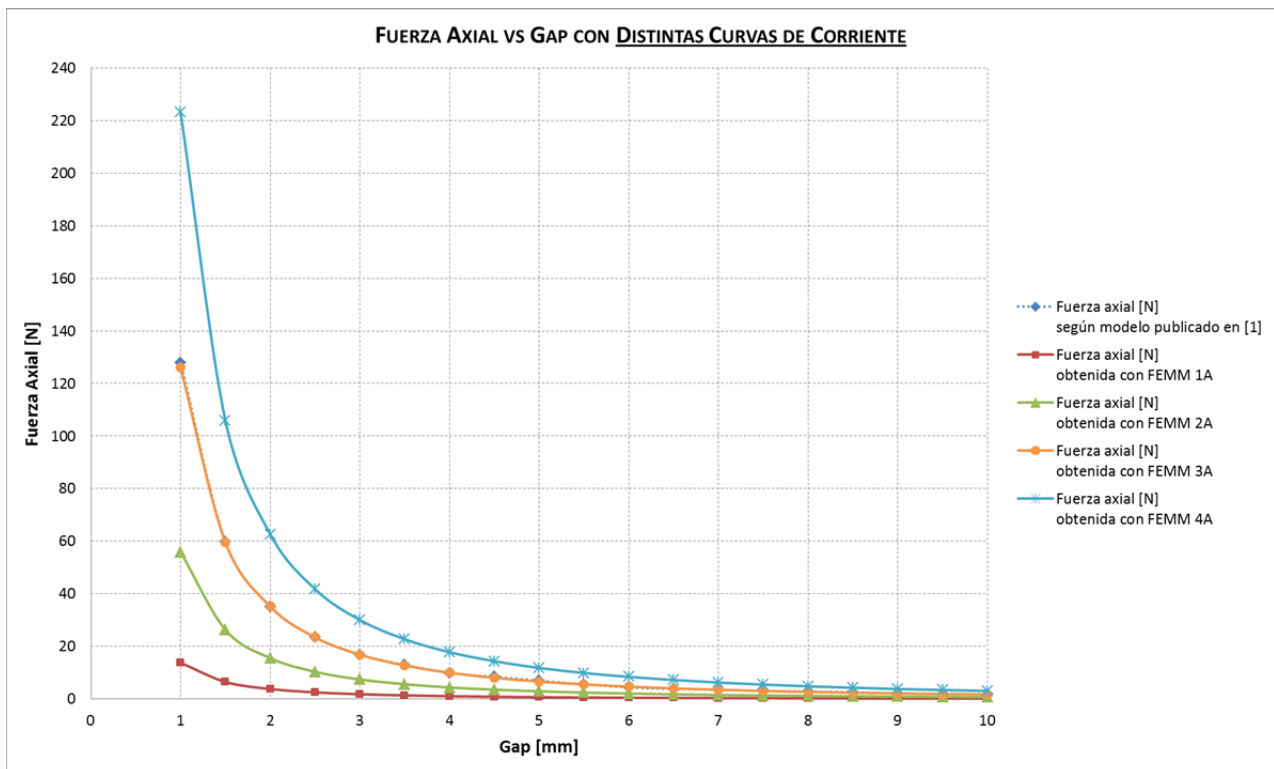
### 7. Representación gráfica de la fuerza axial frente al gap.

Una vez se disponen de todas las medidas de fuerza para los distintos gap, se representa gráficamente la fuerza axial obtenida frente al gap.

### 8. Comparación del modelo simulado con el modelo publicado en [1].

Se va a comparar el modelo simulado con el modelo publicado en [1] para distintos valores de corriente, hasta verificar cuál es el valor de corriente con el que se trabaja en [1]. Puesto que los valores de corriente con los que se trabajan en modelo validado oscilan entre 1-3 A, la simulación se realizará para valores de corriente entre 1-4 A con incrementos de 1 A.

En la Figura 3.6 se representa gráficamente la fuerza axial frente al gap para distintas curvas de corriente. Se observa que la curva del modelo simulado que más se aproxima al modelo publicado en [1] es la obtenida con una corriente DC de 3 A.



**Figura 3.6.** Gráfica de fuerza axial frente al gap con distintas curvas de corriente obtenida con FEMM.

Analizando la gráfica de fuerza axial frente al gap (Figura 3.6), se observa que el comportamiento del modelo simulado se aproxima a lo expuesto en [10], donde se cita que la fuerza axial es inversamente proporcional al cuadrado del gap.

A continuación, para evaluar el comportamiento electromagnético del modelo simulado, se analiza el error relativo medio existente entre la curva de corriente 3 A y la del modelo

publicado en [1] (corriente DC desconocida) por ser la que más se le aproxima, tal y como se ve en la Figura 3.6.

El error relativo se define como un error o incertidumbre en una medición, expresado como la relación del error absoluto y el valor verdadero absoluto. Mientras, el error absoluto se define como la diferencia entre el valor medido, o la aproximación a una cantidad, y el valor verdadero [12].

Así, el error relativo medio se calcula hallando la media aritmética del error relativo, obtenido en %, mediante la expresión (4).

$$E_{relativo} = \left| 1 - \frac{F_{simulacion}}{F_{modelo\ validado}} \right| \cdot 100 \quad (4)$$

Siendo:

$E_{relativo}$ : Error relativo.

$F_{simulacion}$ : Fuerza obtenida en cada gap según el modelo simulado, en Newtons.

$F_{modelo\ validado}$ : Fuerza obtenida en cada gap según el modelo publicado en [1], en Newtons.

Mediante dicha expresión, se ha calculado un error relativo medio del 3,18%, que refleja el error relativo medio obtenido entre la curva de 3 A del modelo simulado y la del modelo publicado en [1] con una corriente constante. Así, se deduce que el modelo simulado con FEMM se ajusta bien al modelo validado, puesto que se ha obtenido un error inferior al 5%. Por tanto, se asume que este primer caso llevado a cabo en el modelo validado se ha realizado con una corriente DC constante de 3 A.

### 3.5 Estudio de fuerza axial frente a la corriente

En este último caso, se va a calcular la fuerza axial ejercida por el estator en el disco rotórico variando la corriente que circula por la bobina del devanado estatórico, manteniendo una distancia de gap constante.

En el modelo publicado en [1] no se especifica el valor constante del gap que se utiliza para este caso. Por este motivo, es necesario realizar varias pruebas para averiguar el valor de gap que se utiliza en el modelo validado.

La corriente inicial será de 0 A y se irá aumentando progresivamente, con incrementos de 0,5 A, hasta alcanzar una corriente de 3 A. Para ello, se irá modificando la corriente que circula por la bobina.

Los tres primeros pasos del procedimiento a seguir en FEMM para este último caso son idénticos a los del caso anterior. Así, el resto de pasos a seguir son:

#### 4. Cálculo de la fuerza axial ejercida por el estator en el rotor frente a la corriente.

Tal y como se realizaba en el caso anterior, una vez iniciado el post-procesador de FEMM, se selecciona el bloque del disco rotórico y se calcula la fuerza axial mediante una integral de fuerza.

#### 5. Aumento de la corriente del devanado estatórico.

Tras disponer del primer valor de fuerza, se incrementa en 0,5 A la corriente que circula por la bobina del devanado estatórico.

#### 6. Repetición de los pasos 1-5.

Se repite la toma de medida de la fuerza axial para cada una de las nuevos valores de corriente DC, hasta alcanzar un corriente de 3 A.

#### 7. Representación gráfica de la fuerza axial frente a la corriente.

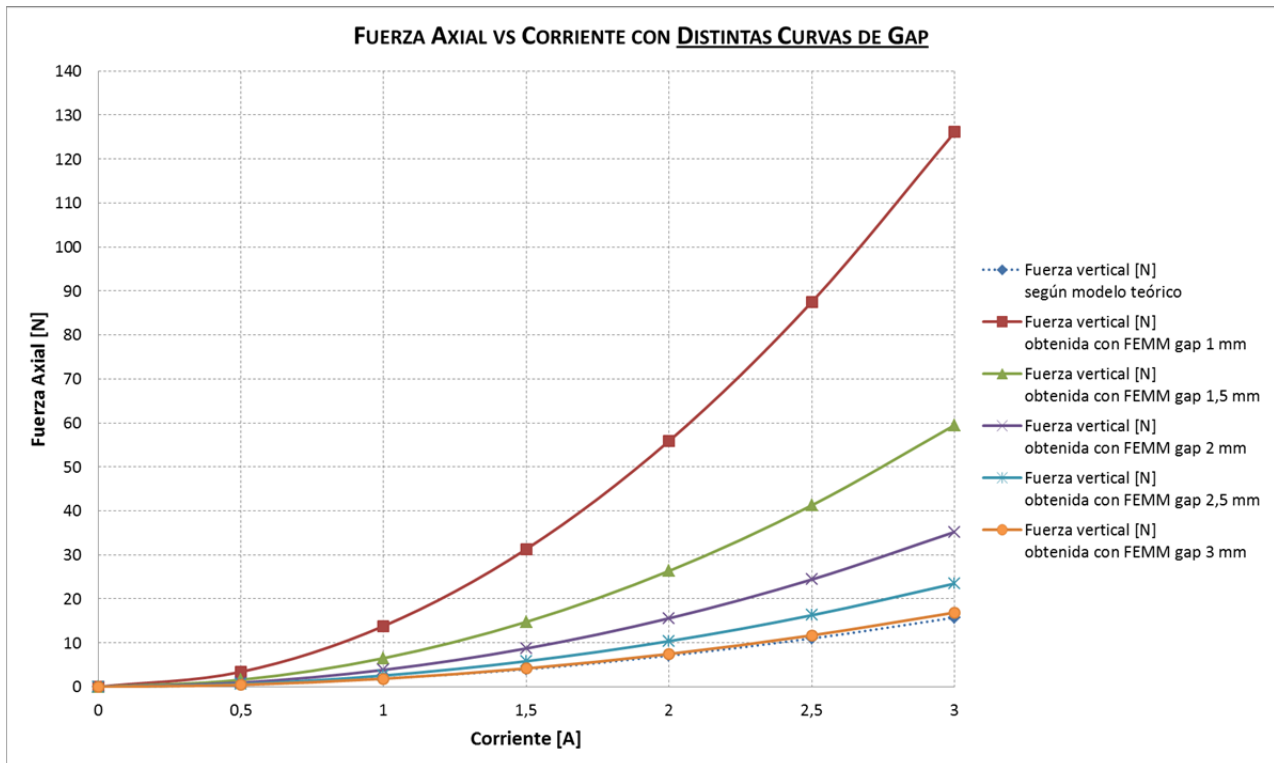
Una vez se disponen de todas las medidas de fuerza para las distintas corrientes, se representa gráficamente la fuerza axial obtenida frente a la corriente.

#### 8. Comparación del modelo simulado con el modelo publicado en [1].

Se va a comparar el modelo simulado con el modelo publicado en [1] para distintos valores de gap, hasta verificar cuál es el valor de gap con el que se trabaja en el modelo validado. Puesto que con valores de gap reducidos se consiguen mayores valores de fuerza magnética, la simulación se realizará para valores de gap bajos. Así, se han realizado varias pruebas con gaps comprendidos entre 1-3 mm con incrementos de 0,5 mm.



En la Figura 3.7 se representa gráficamente la fuerza axial frente a la corriente para distintas curvas de gap. Se observa que la curva del modelo simulado que más se aproxima al modelo publicado en [1] es la obtenida con un gap de 3 mm.



**Figura 3.7.** Gráfica de fuerza axial frente a la corriente con distintas curvas de gap obtenida con FEMM.

Analizando la gráfica de fuerza axial frente a la corriente (Figura 3.7), se observa que el comportamiento del modelo simulado se aproxima a lo expuesto en [10], donde se cita que la fuerza axial es directamente proporcional al cuadrado de la corriente de la bobina que genera el campo magnético.

Análogamente al caso anterior, para evaluar el comportamiento electromagnético del modelo simulado, se analiza el error relativo medio existente entre la curva de gap 3 mm y la del modelo publicado en [1] (gap desconocido) por ser la que más se le aproxima, tal y como se ve en la Figura 3.7.

En este caso, se ha calculado un error relativo medio del 4,94%, que refleja el error relativo medio obtenido entre la curva con gap de 3 mm del modelo simulado y la del modelo publicado en [1] con gap constante. Así, se deduce que el modelo simulado con FEMM se ajusta bien al modelo validado, puesto que se ha obtenido un error inferior al 5%. Por tanto, se asume que este último caso llevado a cabo en el validado se ha realizado con gap constante de 3 mm.

#### **4. METODOLOGÍA EMPLEADA EN EL ESTUDIO PARAMÉTRICO DEL RODAMIENTO MAGNÉTICO**

Tras el análisis de sensibilidad de resultados y de los casos realizados según el modelo publicado en [1], se ha deducido que el primer caso se realiza con una corriente DC de 3 A; y el segundo caso se realiza con un gap de 3 mm. Conocidos estos datos de partida, se puede profundizar en el análisis del resto de parámetros del rodamiento magnético que definen el modelo validado.

Así, se pretende realizar una serie de modificaciones en los parámetros geométricos del rodamiento con la finalidad de obtener un modelo que genere mayor fuerza axial que la del modelo publicado en [1].

Para ello, se ha diseñado un algoritmo con el lenguaje de programación del sistema Matlab. Este algoritmo permite el diseño y análisis de un rodamiento magnético activo axial en FEMM, con las dimensiones que se deseen y siguiendo la estructura descrita anteriormente (Figura 2.6). A continuación, se describen las características de dicho algoritmo programado con Matlab.

La descripción del algoritmo de Matlab se puede dividir en tres partes: la interfaz gráfica, el contenido del algoritmo y los resultados que aporta el mismo.

##### **4.1 Interfaz gráfica del algoritmo de Matlab**

Inicialmente, se había diseñado el algoritmo con el lenguaje de programación del sistema Octave, por ser un *software* de uso libre. No obstante, se descartó esta opción, ya que el paquete de GUI Matlab permite crear una interfaz gráfica mucho más visual y sencilla de comprender que la realizada con Octave.

En las Figuras 4.1 y 4.2, se compara la interfaz gráfica de Octave (Figura 4.1) frente a la creada con Matlab (Figura 4.2) para el algoritmo “FAxial\_vs\_Corriente.m”. En estas figuras se puede observar que la interfaz gráfica de Matlab es mucho más visual, rápida y dinámica que la creada mediante Octave.

Así, la interfaz gráfica creada con Matlab permite:

- **Introducción de parámetros geométricos de una forma más rápida y visual:** Se introducen los parámetros geométricos del rodamiento magnético de una forma mucho más rápida y visual gracias a la imagen del rodamiento que acompaña a la interfaz gráfica y a las indicaciones que acompañan a los parámetros.
- **Validación de errores:** Si alguno de los parámetros es incongruente con la estructura del rodamiento mostrada en la Figura 2.6 (por ejemplo, que la anchura del estator se mayor que su longitud), el programa muestra un mensaje de error (Figura 4.3a).
- **Ejecución de FEMM:** Al pulsar el botón “Aceptar” se muestra un mensaje de texto que resume las dimensiones introducidas del rodamiento. Aquí, nos permite “Ejecutar FEMM” o “Cancelar” para cambiar los parámetros nuevamente (Figura 4.3b).
- **Restablecimiento de parámetros:** Si se desean borrar los parámetros que figuran en la interfaz, se puede pulsar el botón “Restablecer” que reinicia todos los parámetros a 0,00.
- **Cálculo paramétrico del devanado:** Mediante el botón “Cálculo devanado” se calcula el diámetro y sección mínimos que deben tener las espiras y el número de capas necesario según los parámetros introducidos del devanado (corriente y número de espiras) para que la ventana del estator puede albergar la bobina. Para ello, se ha considerado que el conductor debe soportar una densidad de corriente máxima de  $1 \text{ A/mm}^2$  y un factor de empaquetado de 0,9. Además, esta herramienta muestra si el tamaño de la ventana del estator es suficiente para albergar el devanado mediante el mensaje “CORRECTO” o “INCORRECTO” (Figura 4.3c).

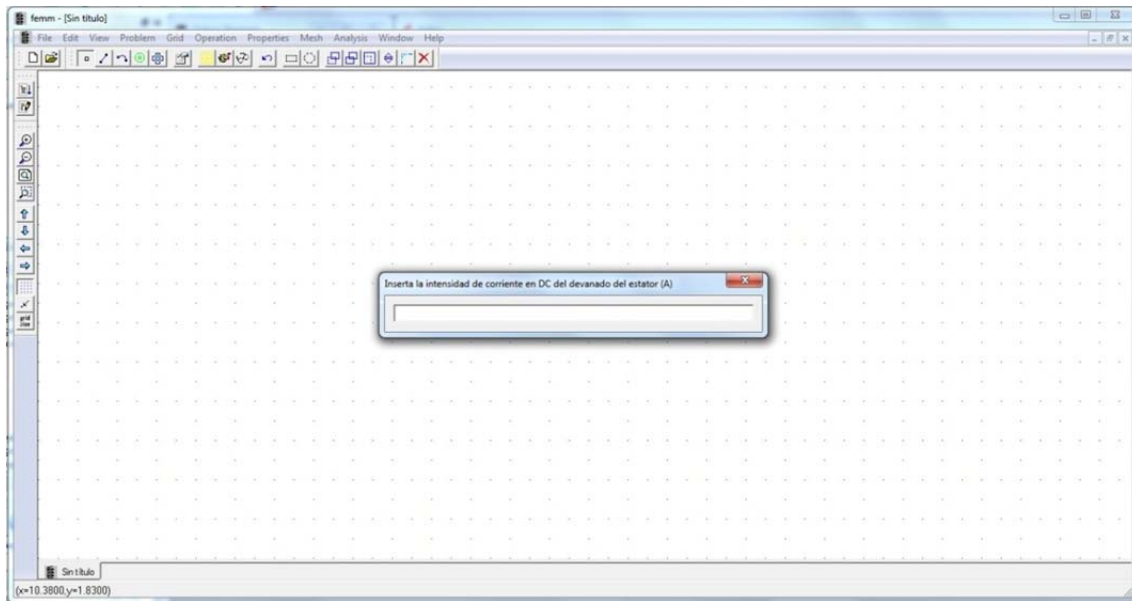


Figura 4.1. Interfaz gráfica creada mediante Octave para el algoritmo "FAxial\_vs\_Corriente.m".

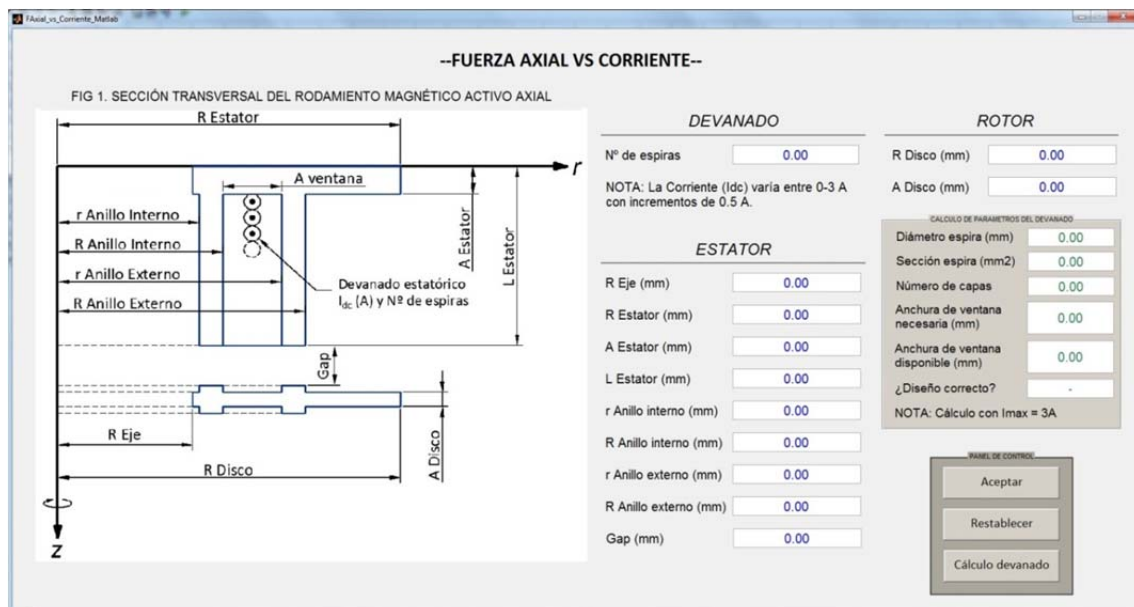


Figura 4.2. Interfaz gráfica creada en Matlab para el algoritmo "FAxial\_vs\_Corriente.m".

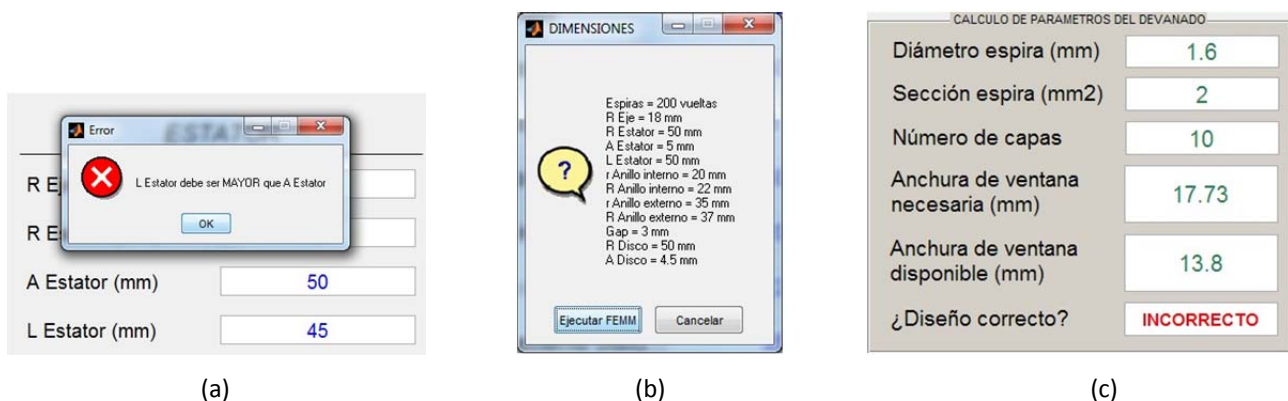


Figura 4.3. Operaciones realizadas mediante la interfaz gráfica del algoritmo "FAxial\_vs\_Corriente.m" mostrando: (a) Validación de errores; (b) Mensaje de ejecución de FEMM; (c) Cálculo de los parámetros del devanado.

## 4.2 Contenido del algoritmo de Matlab

Para hacer un estudio paramétrico del rodamiento magnético más completo, se han diseñado dos algoritmos distintos en Matlab. La finalidad de estos dos algoritmos es implementar los casos que se realizan en el modelo publicado en [1]. Así, cada uno de los algoritmos tiene una función distinta, tal y como se describe a continuación:

- **FAxial\_vs\_Gap.m:** Calcula la fuerza axial ejercida por el estator en el disco rotórico para un valor de corriente DC constante.
- **FAxial\_vs\_Corriente.m:** Calcula la fuerza axial ejercida por el estator en el disco rotórico para un valor de gap constante.

Así, una vez se han introducido los parámetros geométricos del rodamiento magnético mediante la interfaz gráfica de uno de estos dos algoritmos, se ejecuta automáticamente el programa FEMM.

## 4.3 Resultados del algoritmo de Matlab

Tal y como se ha comentado anteriormente, este algoritmo permite diseñar y analizar un rodamiento magnético activo axial utilizando FEMM.

Así, dicho algoritmo nos aporta una serie de datos y herramientas que facilitan el análisis del rodamiento, como son:

- **Almacenamiento en un fichero de FEMM:** El algoritmo crea un fichero de FEMM (RodMag.fem) en el que se guarda el diseño gráfico del rodamiento magnético realizado mediante Matlab.
- **Representación gráfica en Matlab:** Se muestra una gráfica en la que se representa la fuerza axial ejercida por el estator en el rotor frente a la corriente (FAxial\_vs\_Corriente.m) o frente al gap (FAxial\_vs\_Gap.m).
- **Almacenamiento de los datos del gráfico en Matlab:** Se almacenan los datos de los gráficos en ficheros de texto ejecutables en Matlab (valores\_Fvsl.mat y valores\_FvsGap.mat) que pueden ser exportados a otros programas de procesamiento, como pueden ser una hoja de cálculo o de texto.

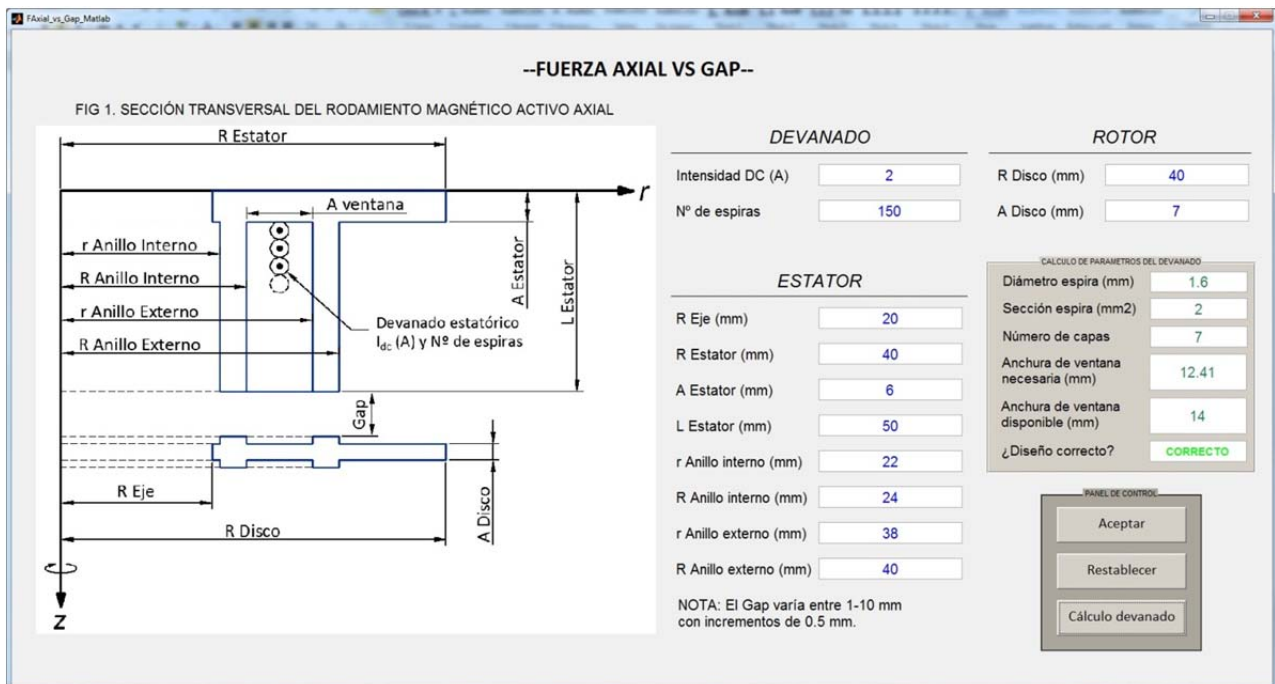
#### 4.4 Ejemplos de funcionamiento de los algoritmos de Matlab

Para aclarar el funcionamiento de estos algoritmos, se va a realizar un ejemplo con el algoritmo "FAXial\_vs\_Gap.m". En la Figura 4.4 se ilustra la interfaz gráfica de Matlab con los parámetros geométricos introducidos.

En la Figura 4.5a se ilustra el cálculo de los parámetros del devanado obtenidos en función del número de espiras y la corriente introducidas. Puesto que el devanado diseñado soporta esa corriente y cabe dentro de la ventana del estator, el diseño es correcto. En la Figura 4.5b se refleja el diseño gráfico de la pieza obtenido con FEMM y almacenado en "RodMag.fem".

En la Figura 4.6a se ilustra la gráfica de la fuerza axial frente al gap obtenida y en la Figura 4.6b los valores de dicho gráfico obtenidos mediante el fichero "valores\_FvsGap.mat".

Finalmente, en la Figura 4.7a se ilustra la gráfica de la fuerza axial frente a la corriente obtenida y en la Figura 4.7b los valores de dicho gráfico obtenidos mediante el fichero "valores\_FvsCorriente.mat" en otro ejemplo.

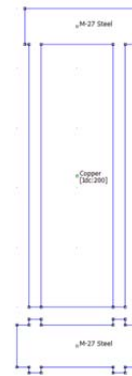


**Figura 4.4.** Interfaz gráfica con los parámetros del ejemplo realizado con el algoritmo "FAXial\_vs\_Gap.m" con una corriente constante de 2 A.

CÁLCULO DE PARÁMETROS DEL DEVANADO

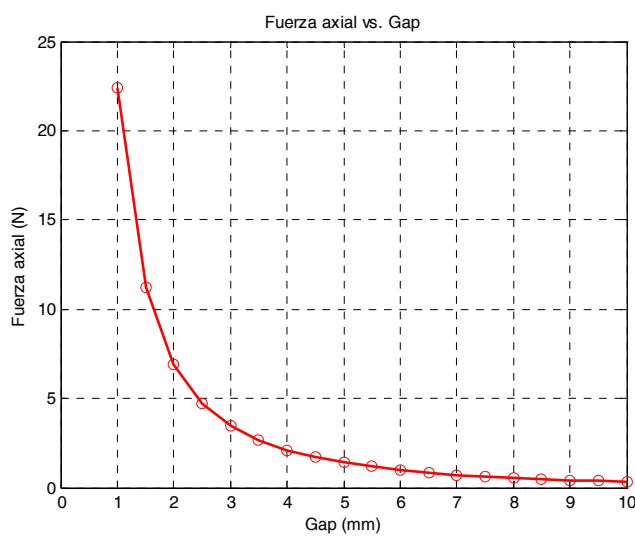
Diámetro espira (mm)	1.6
Sección espira (mm <sup>2</sup> )	2
Número de capas	7
Anchura de ventana necesaria (mm)	12.41
Anchura de ventana disponible (mm)	14
¿Diseño correcto?	CORRECTO

(a)



(b)

**Figura 4.5.** Ejecución de un ejemplo para el algoritmo "FAxial\_vs\_Gap.m" con una corriente de 2 A mostrando: (a) Cálculo de los parámetros del devanado; (b) Diseño gráfico obtenido con FEMM.



(a)

Command Window

New to MATLAB? Watch this [Video](#).

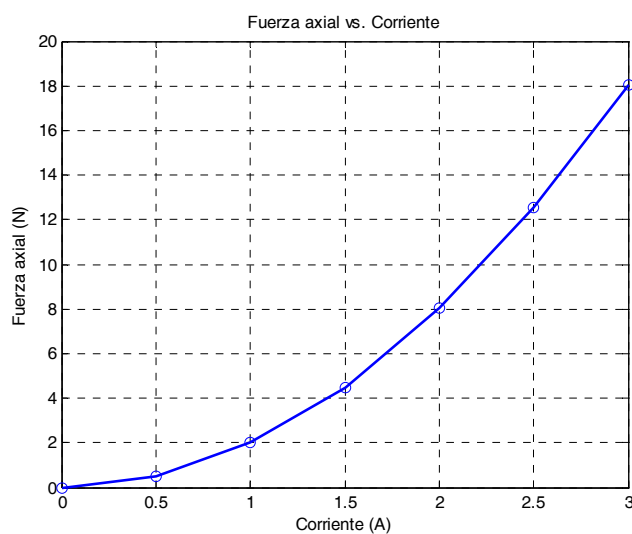
```
>> [x f]
```

ans =

1.0000	22.4226
1.5000	11.2136
2.0000	6.8764
2.5000	4.7079
3.0000	3.4514
3.5000	2.6376
4.0000	2.0926
4.5000	1.6900
5.0000	1.3932
5.5000	1.1720
6.0000	0.9904
6.5000	0.8459
7.0000	0.7288
7.5000	0.6344
8.0000	0.5535
8.5000	0.4893
9.0000	0.4337
9.5000	0.3829
10.0000	0.3421

(b)

**Figura 4.6.** Ejecución de un ejemplo para el algoritmo "FAxial\_vs\_Gap.m" con una corriente de 2 A mostrando: (a) Gráfica de la fuerza axial frente al gap; (b) Valores del gráfico de fuerza axial frente al gap.



(a)

Command Window

New to MATLAB? Watch this [Video](#).

```
>> [x f]
```

ans =

0	0
0.5000	0.4902
1.0000	1.9884
1.5000	4.5024
2.0000	8.0289
2.5000	12.5576
3.0000	18.0580

(b)

**Figura 4.7.** Ejecución de un 2º ejemplo para el algoritmo "FAxial\_vs\_Corriente.m" con gap de 2 mm: (a) Gráfica de la fuerza axial frente a la corriente; (b) Valores del gráfico de fuerza axial frente a la corriente.

## 5. RESULTADOS DEL ESTUDIO PARAMÉTRICO DEL RODAMIENTO MAGNÉTICO

Una vez se dispone del algoritmo de Matlab que nos permite modificar las dimensiones y parámetros del rodamiento magnético de una forma más dinámica, se pueden realizar las pruebas necesarias para estudiar el comportamiento del mismo.

A continuación, se describen las distintas pruebas que se han realizado en el rodamiento cuya finalidad es simular un modelo que aporte más fuerza axial que el modelo publicado en [1].

Para ello, en cada prueba se trata de variar un único parámetro manteniendo constante el resto de valores descritos en la Tabla 2.1.

### 5.1 Modificación de los diámetros

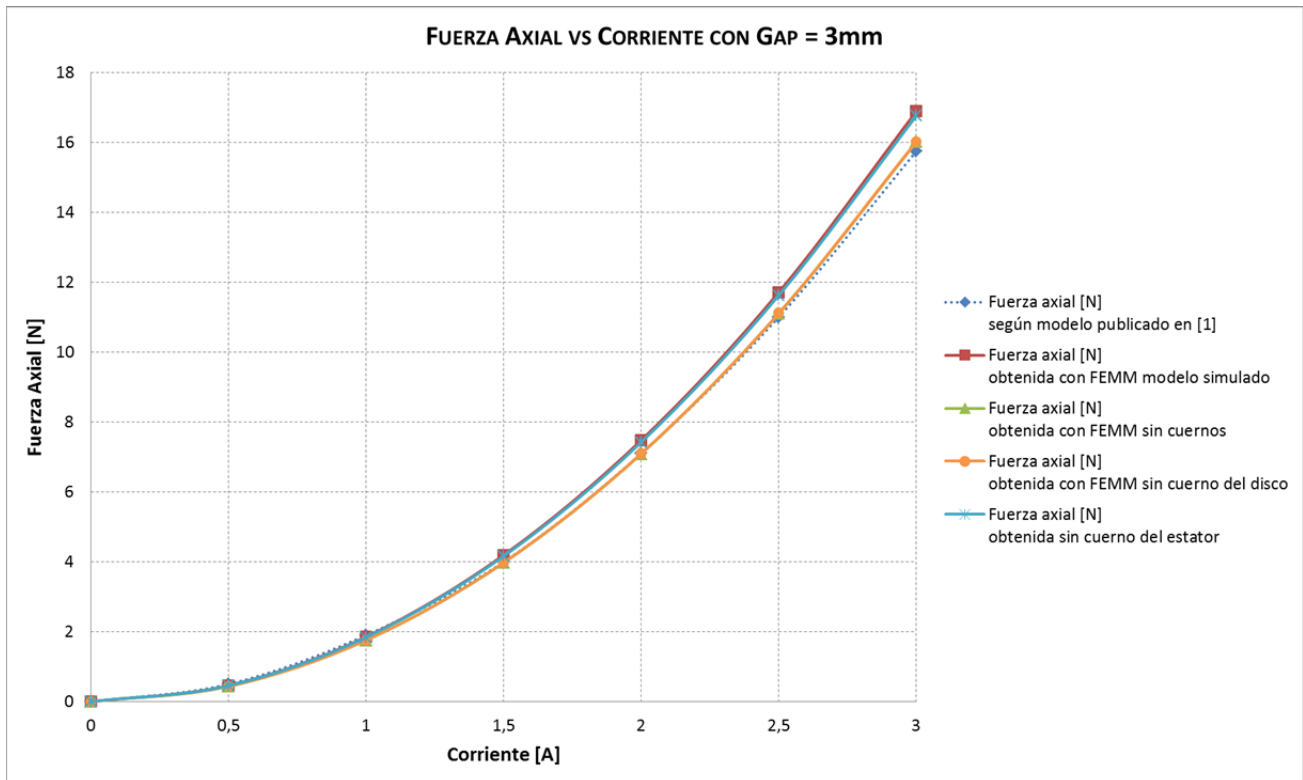
En primer lugar, se han variado los diámetros del rodamiento para evaluar su comportamiento electromagnético.

A continuación, se ilustra la gráfica de la fuerza axial frente a la corriente con un gap constante de 3 mm (Figura 5.1) y frente al gap con una corriente DC constante de 3 A (Figura 5.2), con varias modificaciones en los diámetros del rodamiento.

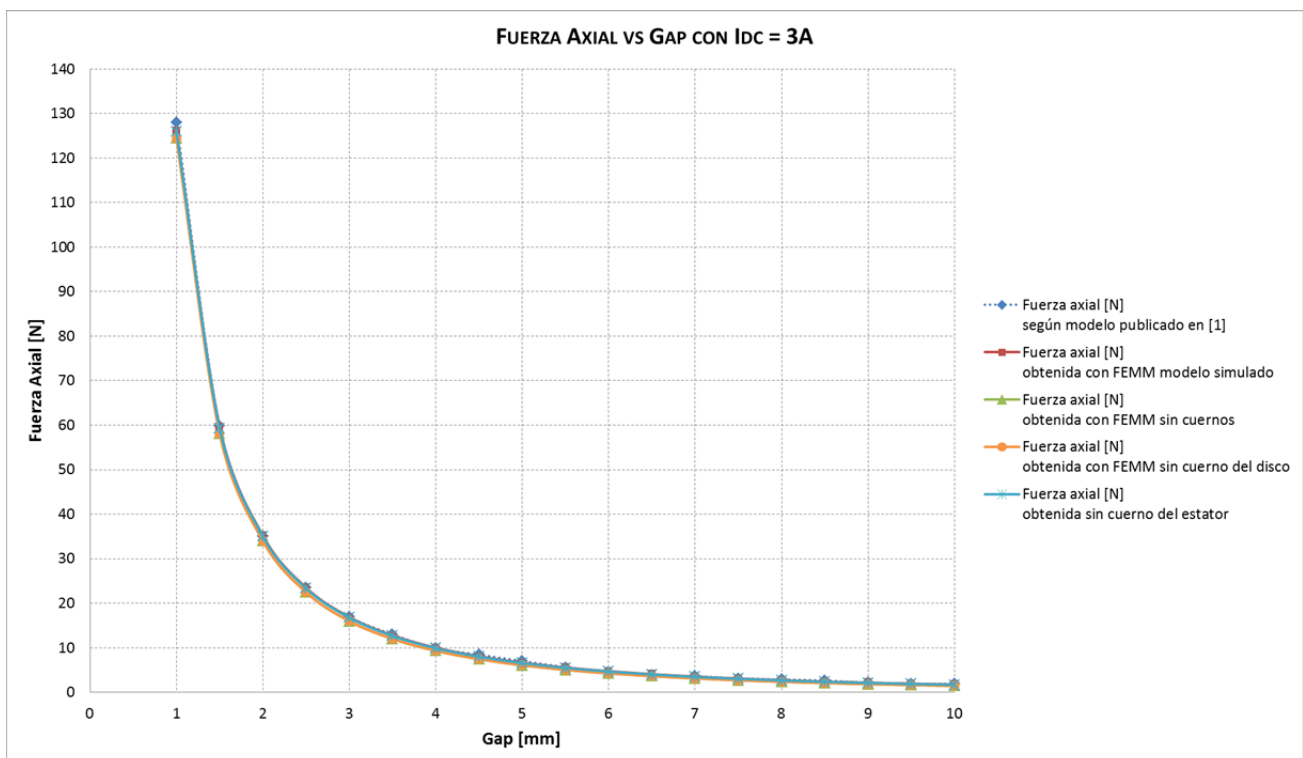
Además, en la Figura 5.3 se ilustran las modificaciones aplicadas a los diámetros del rodamiento vistas en FEMM. Estas modificaciones consisten en lo siguiente:

- **Modelo sin cuernos en el rodamiento:** En este modelo se eliminan los cuernos del estator y del rotor, haciendo que sus nuevas medidas coincidan con el valor del radio exterior del anillo electromagnético externo. (Figura 5.3a).
- **Modelo sin cuerno del disco:** En este modelo se elimina el cuerno del rotor, haciendo que su nueva medida coincida con el valor del radio exterior del anillo electromagnético externo (Figura 5.3b).
- **Modelo sin cuerno del estator:** En este modelo se elimina el cuerno del estator, haciendo que su nueva medida coincida con el valor del radio exterior del anillo electromagnético externo (Figura 5.3c).

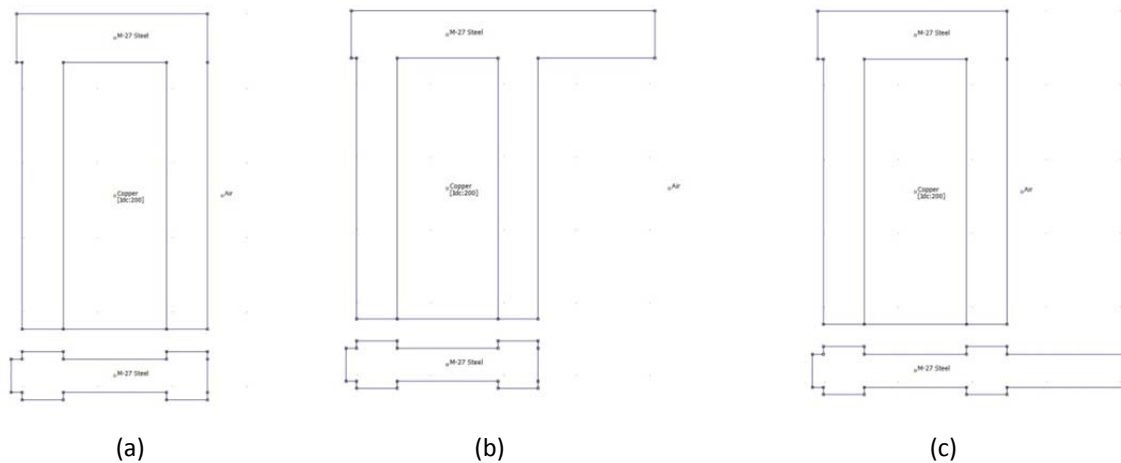




**Figura 5.1.** Fuerza axial frente a la corriente con un gap de 3 mm para rodamientos con modificaciones en los diámetros.



**Figura 5.2.** Fuerza axial frente al gap con un corriente DC de 3 A para rodamientos con modificaciones en los diámetros.



**Figura 5.3.** Diseño gráfico del rodamiento magnético obtenido en FEMM con el algoritmo "FAxial\_vs\_Corriente.m" (a) sin cuernos en el rodamiento; (b) sin cuerno del disco rotórico y (c) sin cuerno del disco del estator.

Atendiendo a las gráficas de la fuerza axial frente a la corriente (Figura 5.1) y frente al gap (Figura 5.2), se aprecia que las curvas obtenidas con las modificaciones de los diámetros son muy similares a la curva del modelo publicado en [1].

Por tanto, al modificar los diámetros del rodamiento del modelo publicado en [1] no se consiguen mayores valores de fuerza axial.

## 5.2 Modificación de anchuras

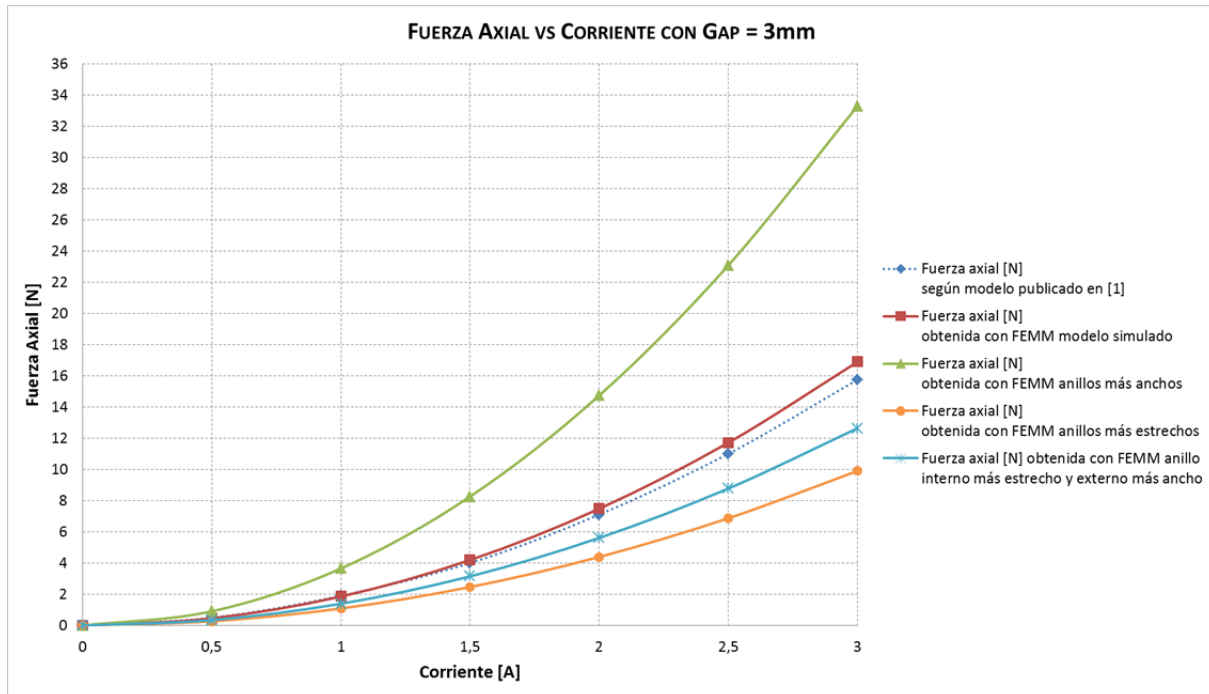
En segundo lugar, se han variado las anchuras del rodamiento para evaluar su comportamiento.

A continuación, se ilustra la gráfica de la fuerza axial frente a la corriente con un gap constante de 3 mm (Figura 5.4) y frente al gap con una corriente DC constante de 3 A (Figura 5.5), con varias modificaciones en las anchuras del rodamiento.

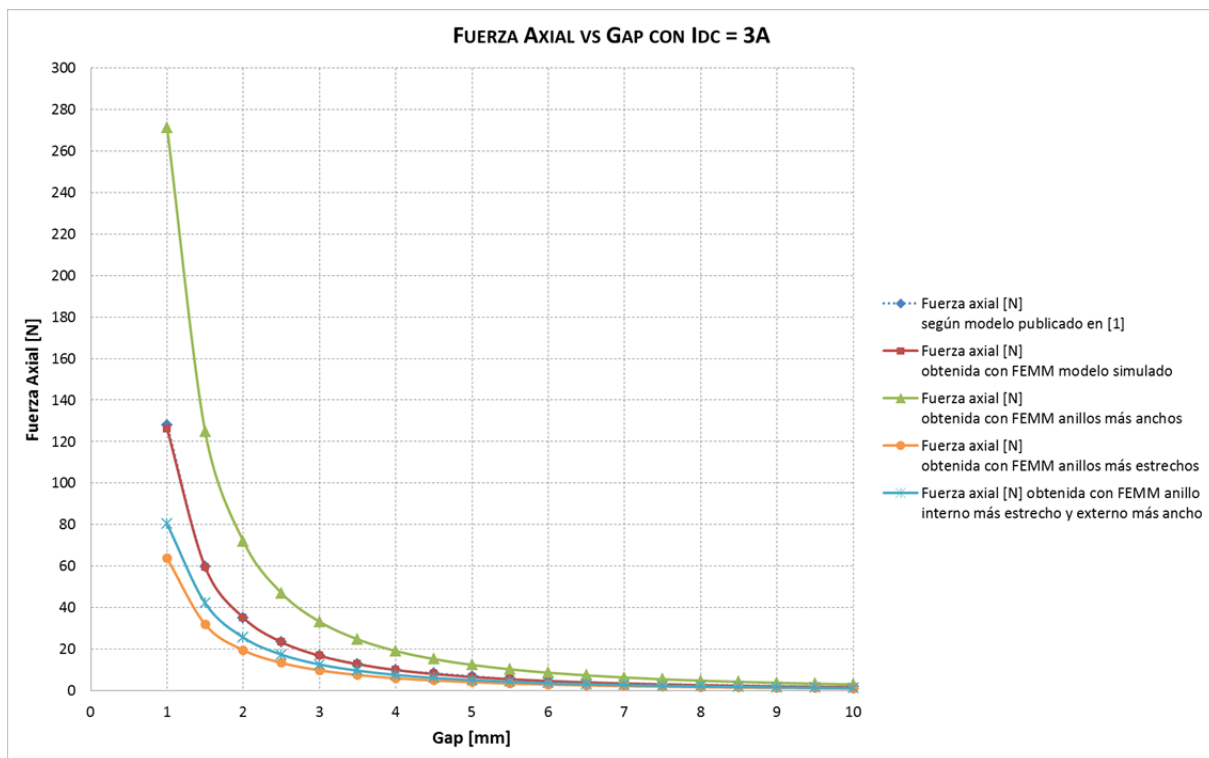
Además, en la Figura 5.6 se ilustran las modificaciones aplicadas a las anchuras del rodamiento vistas en FEMM. Estas modificaciones consisten en lo siguiente:

- **Modelo con anillos electromagnéticos más anchos:** En este modelo se aumenta la anchura de los anillos electromagnéticos interno y externo al doble del tamaño descrito en el modelo publicado en [1], que en este caso es de 5,5 mm. Por tanto, la nueva anchura de los anillos es de 10 mm (Figura 5.6a).
- **Modelo con anillos electromagnéticos más estrechos:** En este modelo se reduce la anchura de los anillos electromagnéticos interno y externo a la mitad del tamaño descrito en el modelo publicado en [1]. Por tanto, la nueva anchura de los anillos es de 2,75 mm (Figura 5.6b).

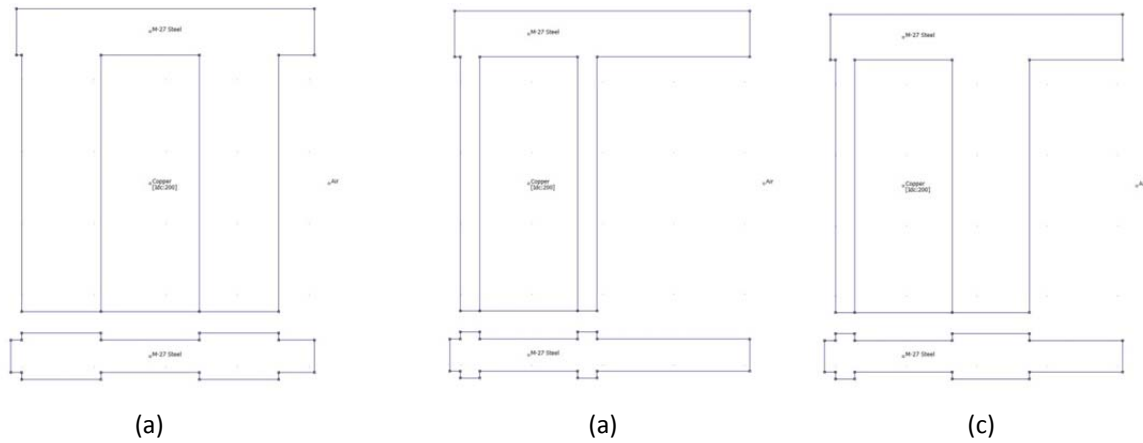
- **Modelo con anillos interno más estrecho y externo más ancho:** En este modelo se reduce la anchura del anillo interno a la mitad (2,75 mm) y se aumenta la anchura del anillo externo al doble (10 mm), según los datos descritos en el modelo publicado en [1] (Figura 5.6c).



**Figura 5.4.** Fuerza axial frente a la corriente con un gap de 3 mm para rodamientos con modificaciones en las anchuras.



**Figura 5.5.** Fuerza axial frente al gap con un corriente DC de 3 A para rodamientos con modificaciones en las anchuras.



**Figura 5.6.** Diseño gráfico del rodamiento magnético obtenido en FEMM con el algoritmo “FAxial\_vs\_Corriente.m” (a) con anillos más gruesos, (b) con anillos más estrechos y (c) con anillo interno más estrecho y anillo externo más ancho.

Atendiendo a la gráfica de la fuerza axial frente a la corriente (Figura 5.4), se observa que con la curva obtenida con anillos electromagnéticos el doble de anchos que los del modelo publicado en [1], se obtienen valores de fuerza de axial superiores a los resultados del modelo validado. El resto de las curvas obtenidas dan lugar a valores de fuerza inferiores.

Del mismo modo, atendiendo a la gráfica de la fuerza axial frente al gap (Figura 5.5), se aprecia que con la curva obtenida con anillos electromagnéticos el doble de anchos que los del modelo publicado en [1], se obtienen valores de fuerza axial superiores a los resultados del modelo validado.

Por tanto, al modificar el parámetro de la anchura del rodamiento se observan importantes variaciones en el comportamiento del mismo. Así, con un rodamiento magnético con anillos electromagnéticos el doble de anchos que los del modelo publicado en [1] se obtienen valores de fuerza axial superiores a los obtenidos en el modelo validado.

Los parámetros empleados en el diseño de dicho modelo del rodamiento con anillos electromagnéticos el doble de anchos que los del modelo teórico (10 mm), se ilustran en la Tabla 5.1.

**TABLA 5.1.** PARÁMETROS DEL MODELO DEL RODAMIENTO MAGNÉTICO CON ANILLOS ELECTROMAGNÉTICOS DE 10 mm DE ANCHURA.

PARÁMETRO	VALOR
Número de espiras del devanado	200 vueltas
Diámetro de la espira	1 mm
Radio del Eje	18,4 mm
Radio del Estator	60,7 mm
Anchura del Estator	6,5 mm
Longitud del Estator	42,3 mm
Radio Interior del Anillo Interno	19,9 mm
Radio Exterior del Anillo Interno	30,9 mm
Radio Interior del Anillo Externo	44,7 mm
Radio Exterior del Anillo Externo	55,7 mm
Radio del Disco Rotórico	60,7 mm
Anchura del Disco Rotórico	4,5 mm
Corriente DC (Caso 3.4)	3 A
Gap (Caso 3.5)	3 mm

No obstante, este aumento de fuerza se consigue a costa de un incremento en la masa del disco rotórico. Así, dicha masa se puede calcular a partir de la siguiente expresión (5).

$$d_{acero} = \frac{m_{disco}}{V_{disco}} \Rightarrow m_{disco} = V_{disco} \cdot d_{acero} \quad (5)$$

Siendo:

$d_{acero}$ : Densidad del acero, se toma el valor de 7850 kg/m<sup>3</sup>.

$m_{disco}$ : Masa del disco rotórico, en kg.

$V_{disco}$ : Volumen del disco rotórico, en m<sup>3</sup>.

En la Tabla 5.2 se reflejan los valores de masa del disco rotórico obtenidos según el modelo publicado en [1] y según el modelo simulado con anillos electromagnéticos el doble de anchos (10 mm) que los del modelo validado. Se puede apreciar que los resultados de fuerza axial obtenidos se consiguen a costa de incrementar la masa del disco rotórico en un 23,50%.

**TABLA 5.2.** MASA DEL DISCO ROTÓRICO SEGÚN EL MODELO PUBLICADO EN [1]  
Y SEGÚN EL MODELO SIMULADO CON ANILLOS ELECTROMAGNÉTICOS DE 10 mm.

PARÁMETRO	VALOR
Masa del disco según el modelo publicado en [1]	100,04 gr
Masa del disco según el modelo simulado con anillos más anchos	123,55 gr
Incremento de masa del disco	23,50%

En la Tabla 5.3 se ilustra una comparación entre la fuerza axial obtenida según el modelo publicado en [1] y el incremento de fuerza obtenido con el modelo simulado con los anillos electromagnéticos más anchos mediante el algoritmo “FAxial\_vs\_Gap.m”.

**TABLA 5.3.** COMPARACIÓN DE FUERZA AXIAL OBTENIDA FRENTE AL GAP ENTRE EL MODELO PUBLICADO EN [1]  
Y EL MODELO SIMULADO CON ANILLOS DE 10 mm CON UNA CORRIENTE CONSTANTE DC DE 3 A.

GAP [MM]	FUERZA AXIAL SEGÚN EL MODELO VALIDADO [N]	FUERZA AXIAL SEGÚN EL MODELO CON ANILLOS MÁS ANCHOS [N]	INCREMENTO DE FUERZA OBTENIDO [%]
1	128,00	271,07	111,77
1,5	59,85	124,68	108,32
2	35,11	72,04	105,17
2,5	23,46	47,09	100,72
3	16,94	33,28	96,47
3,5	13,00	24,80	90,75
4	9,94	19,19	93,07
4,5	8,48	15,29	80,35
5	7,03	12,48	77,49
5,5	5,65	10,35	83,25
6	4,41	8,72	97,83
6,5	3,83	7,45	94,54
7	3,63	6,42	76,89
7,5	3,15	5,57	76,95
8	2,91	4,89	68,07
8,5	2,67	4,32	61,96
9	2,18	3,83	75,81
9,5	1,94	3,42	76,33
10	1,94	3,07	58,42

En la Tabla 5.4 se ilustra una comparación entre la fuerza axial obtenida con el modelo publicado en [1] y el incremento de fuerza obtenido con el modelo simulado con anillos electromagnéticos de 10 mm mediante el algoritmo “FAxial\_vs\_Corriente.m”.

**TABLA 5.4.** COMPARACIÓN DE FUERZA AXIAL OBTENIDA FRENTE A LA CORRIENTE ENTRE EL MODELO PUBLICADO EN [1] Y EL MODELO SIMULADO CON ANILLOS DE 10 mm CON UN GAP CONSTANTE DE 3 mm.

CORRIENTE [A]	FUERZA AXIAL SEGÚN EL MODELO VALIDADO[8] [N]	FUERZA AXIAL SEGÚN EL MODELO CON ANILLOS MÁS ANCHOS [N]	INCREMENTO DE FUERZA OBTENIDO [%]
0	0,00	0,00	0,00
0,5	0,50	0,91	81,00
1	1,90	3,65	92,26
1,5	4,00	8,26	106,55
2	7,10	14,74	107,57
2,5	11,00	23,08	109,81
3	15,75	33,29	111,34

Este aumento de fuerza axial se debe a que, atendiendo a la resolución de circuitos magnéticos, un aumento de la sección del circuito magnético da lugar a una disminución en la reluctancia del estator y esto, a su vez, provoca que aumente la fuerza axial ejercida por éste en el rotor.

## 6. DISEÑO DE UN RODAMIENTO MAGNÉTICO ACTIVO AXIAL PARA UN VOLANTE DE INERCIA

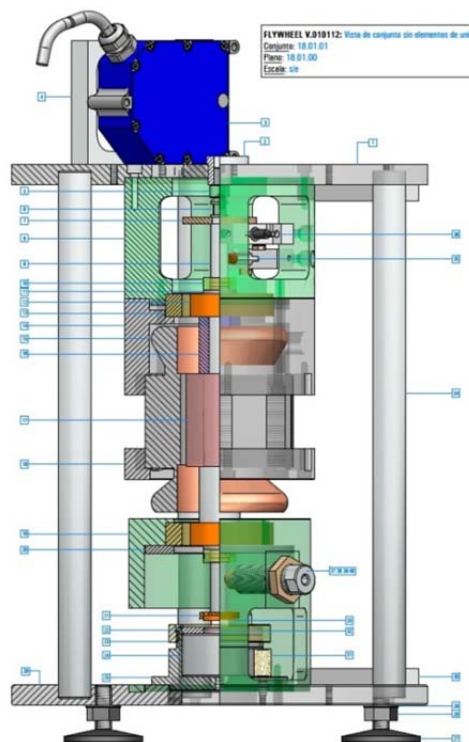
Los volantes de inercia, como sistemas de almacenamiento de energía, se basan en el uso de una masa rotativa (rotor) que gira a alta velocidad para almacenar energía cinética. Esta energía se puede extraer o inyectar mediante un generador o un motor, respectivamente [13]. Para garantizar una elevada eficiencia energética, los volantes de inercia trabajan dentro de una cámara de vacío. Por este motivo, el uso de rodamientos magnéticos es ideal para este tipo de sistema de almacenamiento de energía.

Las ventajas de los volantes de inercia frente a otros sistemas de almacenamiento convencionales, como las baterías electroquímicas, consisten en que son menos contaminantes y aseguran una mayor eficiencia energética [4]. Además, aportan una gran densidad de potencia de salida y mayor vida útil [13].

Tras esto, se describe el diseño de un rodamiento magnético axial del tipo estudiado en este TFM para el prototipo construido en [8], cuyas características se describen a continuación.

### 6.1 Descripción del prototipo operativo de rotor levitado

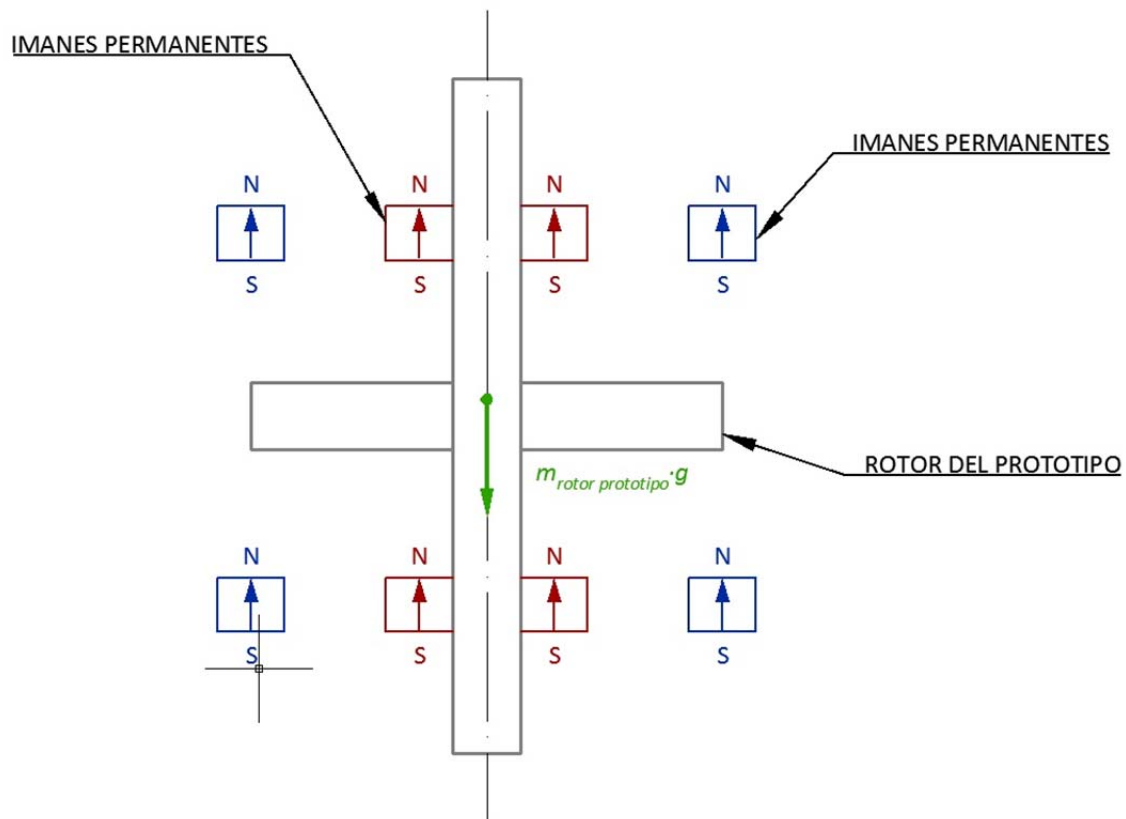
El prototipo que se describe, se basa en un rodamiento magnético pasivo radial, que estabiliza el movimiento del rotor en dirección radial mediante imanes permanentes. Su estructura general se ilustra en la Figura 6.1.



**Figura 6.1.** Estructura del prototipo operativo de rotor levitado de [8].



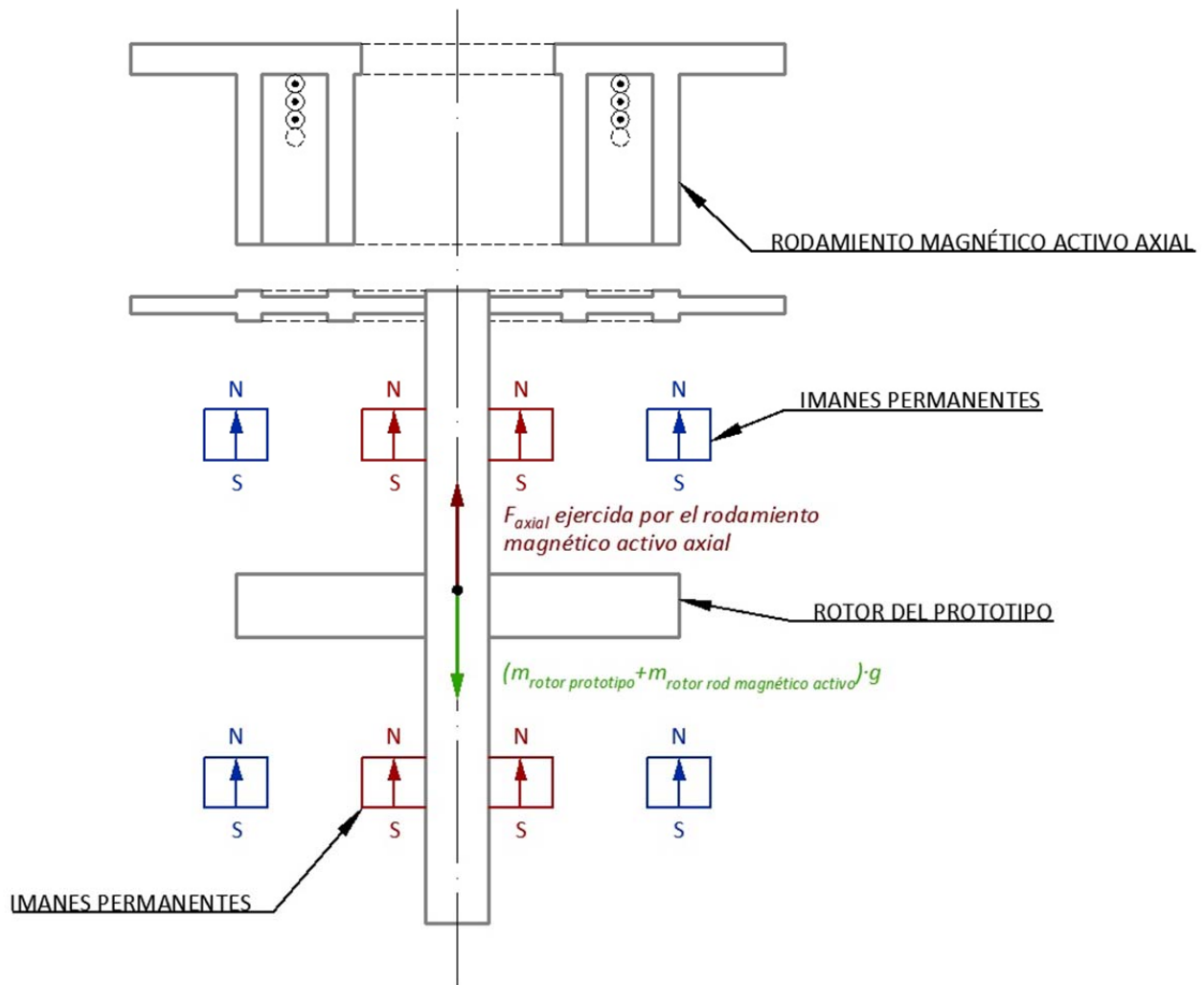
En la Figura 6.2 se ilustra una vista muy básica de la estructura del rotor del prototipo al aplicar una sección transversal. En ella, se puede apreciar que el movimiento del rotor en dirección radial se compensa gracias a dos imanes de la misma dirección de magnetización que se encuentran enfrentados lateralmente. Así, cuando los imanes de la estructura del rotor (rojo) se acercan radialmente a los imanes exteriores (azul) son repelidos por éstos. Por otro lado, se aprecia que la fuerza en dirección axial que se pretende compensar corresponde al peso de la estructura del rotor ( $m_{\text{rotor prototipo}} \cdot g$ ).



**Figura 6.2.** Sección transversal de la estructura del rotor del prototipo operativo de rotor levitado realizada con Autocad.

Por tanto, con este estudio se pretende diseñar un rodamiento magnético que compense las fuerzas en dirección axial, es decir, que compense el peso del rotor con la fuerza axial ejercida por el rodamiento, gracias al trabajo previo realizado en este TFM.

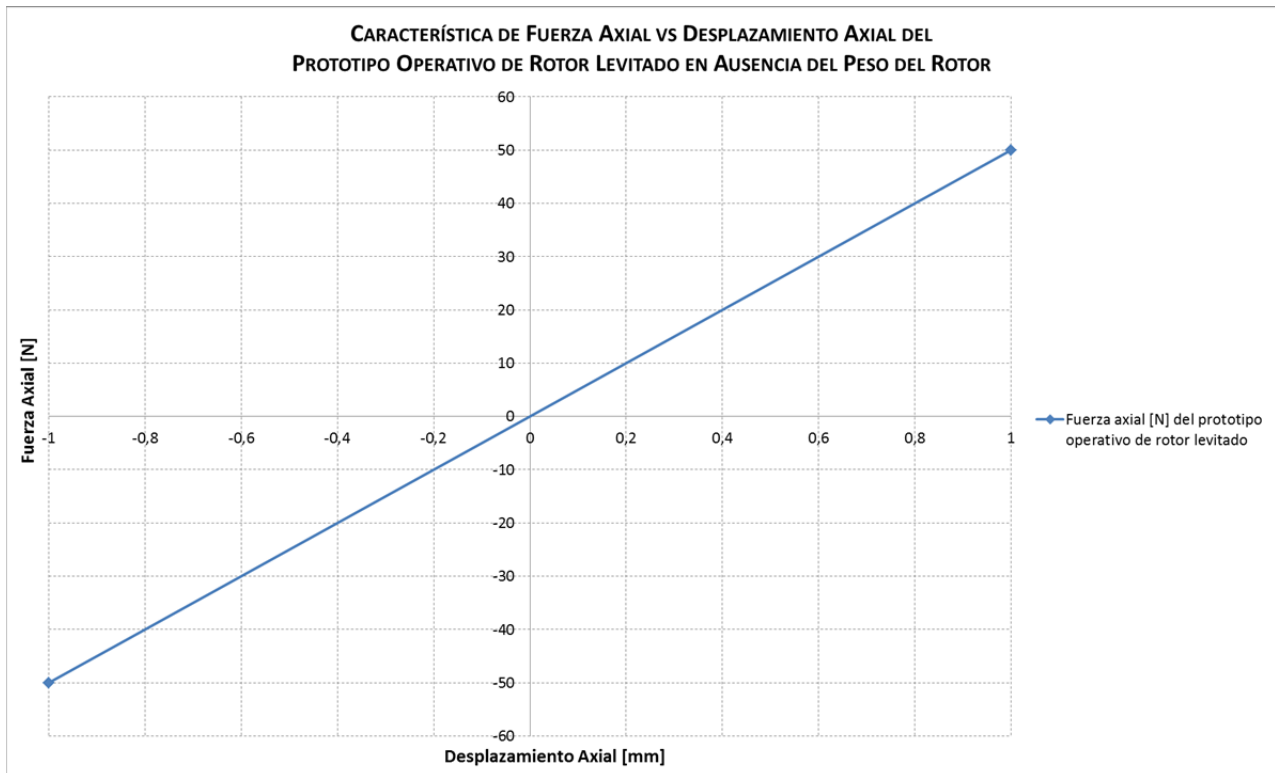
En la Figura 6.3 se ilustra el acoplamiento del rodamiento magnético activo axial que se debería diseñar para compensar la fuerza en dirección axial del prototipo de [8].



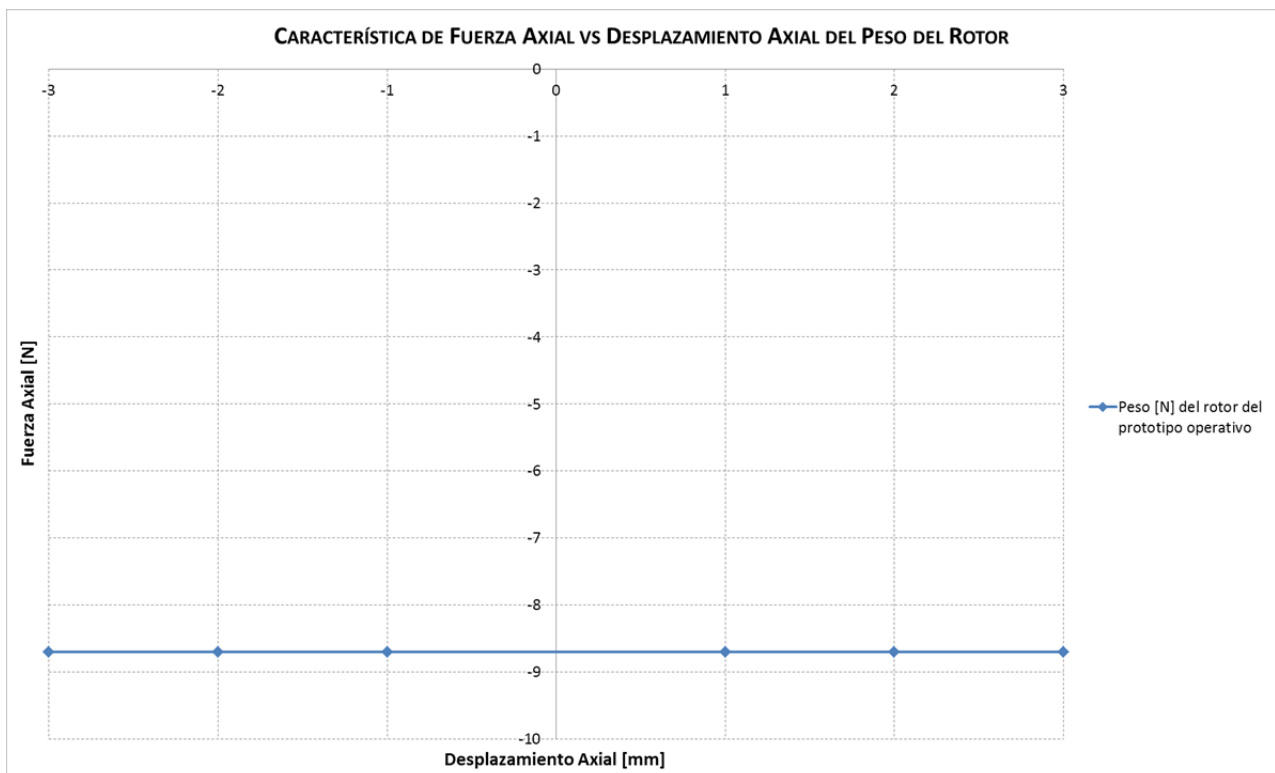
**Figura 6.3.** Acoplamiento del rodamiento magnético activo axial al prototipo operativo de rotor levitado.

Para diseñar un modelo adecuado, hay que tener en cuenta que si el rotor se desplaza en dirección axial (por ejemplo, hacia arriba), el polo Norte del imán externo atraerá al polo Sur del imán acoplado a la estructura del rotor, creando una fuerza axial a considerar en el diseño del rodamiento. Así, hay que caracterizar esa fuerza en función del desplazamiento axial. Para ello, se ha tomado como origen del desplazamiento axial el punto en el que los imanes externos están alineados con los imanes de la estructura del rotor.

Dicha característica de fuerza axial frente al desplazamiento axial se ha obtenido de los datos facilitados por [8]. Así, en la Figura 6.4 se aprecia que la característica de fuerza axial frente al desplazamiento axial tiene un comportamiento lineal de pendiente 50 N/mm. Del mismo modo, se puede representar el peso del rotor (8,7 N [8]) frente al desplazamiento, tal y como se ilustra en la Figura 6.5.

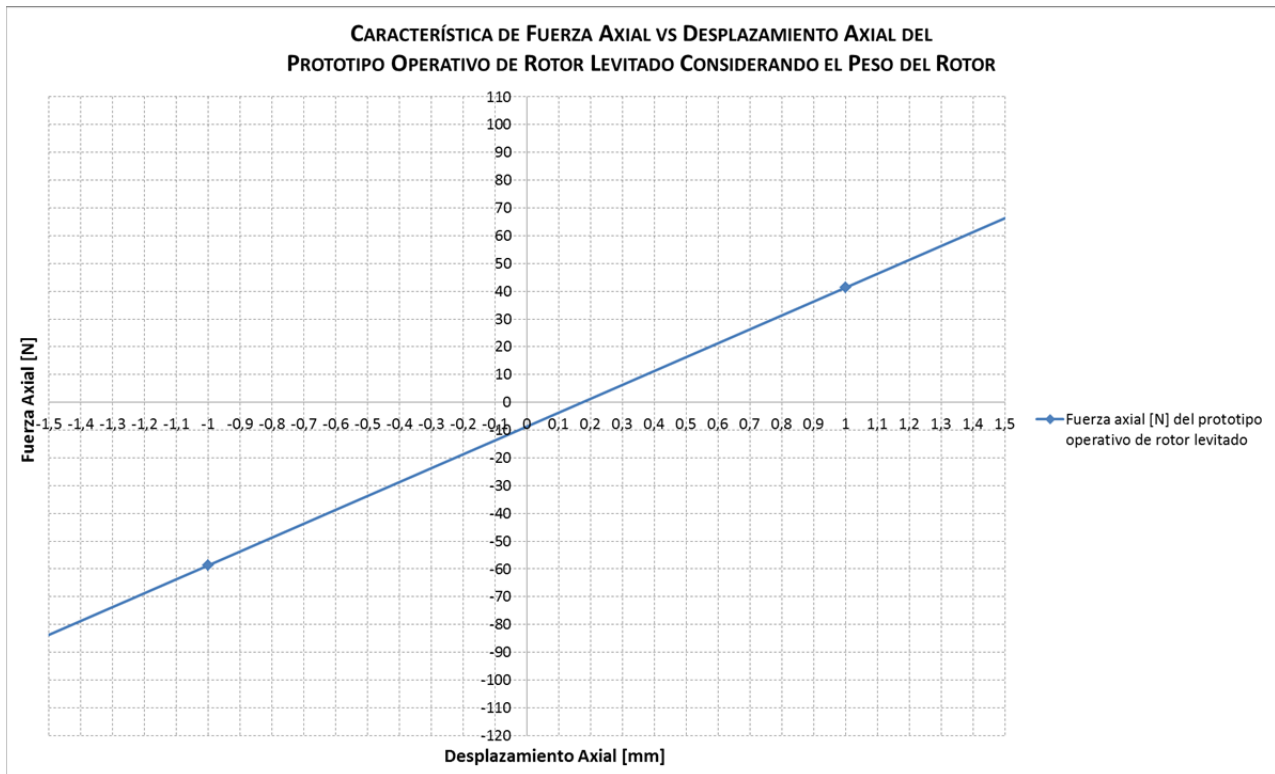


**Figura 6.4.** Característica de fuerza axial frente al desplazamiento axial del prototipo de rotor levitado en ausencia del peso del rotor.



**Figura 6.5.** Gráfica del peso del rotor del prototipo de rotor levitado frente al desplazamiento axial.

Al juntar ambas gráficas, la característica de fuerza frente al desplazamiento axial queda desplazada verticalmente -8,7 N, debido al peso del rotor. En la Figura 6.6 se puede apreciar el desplazamiento de dicha gráfica.



**Figura 6.6.** Característica de fuerza axial frente al desplazamiento axial del prototipo de rotor levitado en considerando el peso del rotor.

De esta figura se deduce que el punto en el que la fuerza axial es nula y, por tanto, el punto en el que el rotor está en equilibrio axial se consigue para un desplazamiento de 0,15 mm. A partir de este desplazamiento de 0,15, se entra en la zona de inestabilidad no controlable, por tanto habrá que trabajar en el cuarto cuadrante.

Se va a diseñar un rodamiento que garantice, al menos, el doble de la fuerza axial obtenida con un desplazamiento axial de -1,5 mm, que en este caso es de -87 N. Por tanto, el modelo simulado de rodamiento debe aportar, al menos, -174 N para operar en torno al punto de equilibrio, que se sitúa en -1 mm respecto al origen del desplazamiento axial. Asimismo, para trabajar en torno a dicho punto de equilibrio, se debe considerar una amplitud máxima de movimiento axial comprendida entre -0,5 y -1,5 mm.

## 6.2 Estudio a realizar en el prototipo operativo de rotor levitado

En primer lugar, se deben considerar los parámetros geométricos iniciales del prototipo antes de realizar cualquier tipo de ensayo. Así, los parámetros del rotor del prototipo se ilustran en la Tabla 6.1 [8].

**TABLA 6.1.** PARÁMETROS DEL ROTOR DEL PROTOTIPO DE [8].

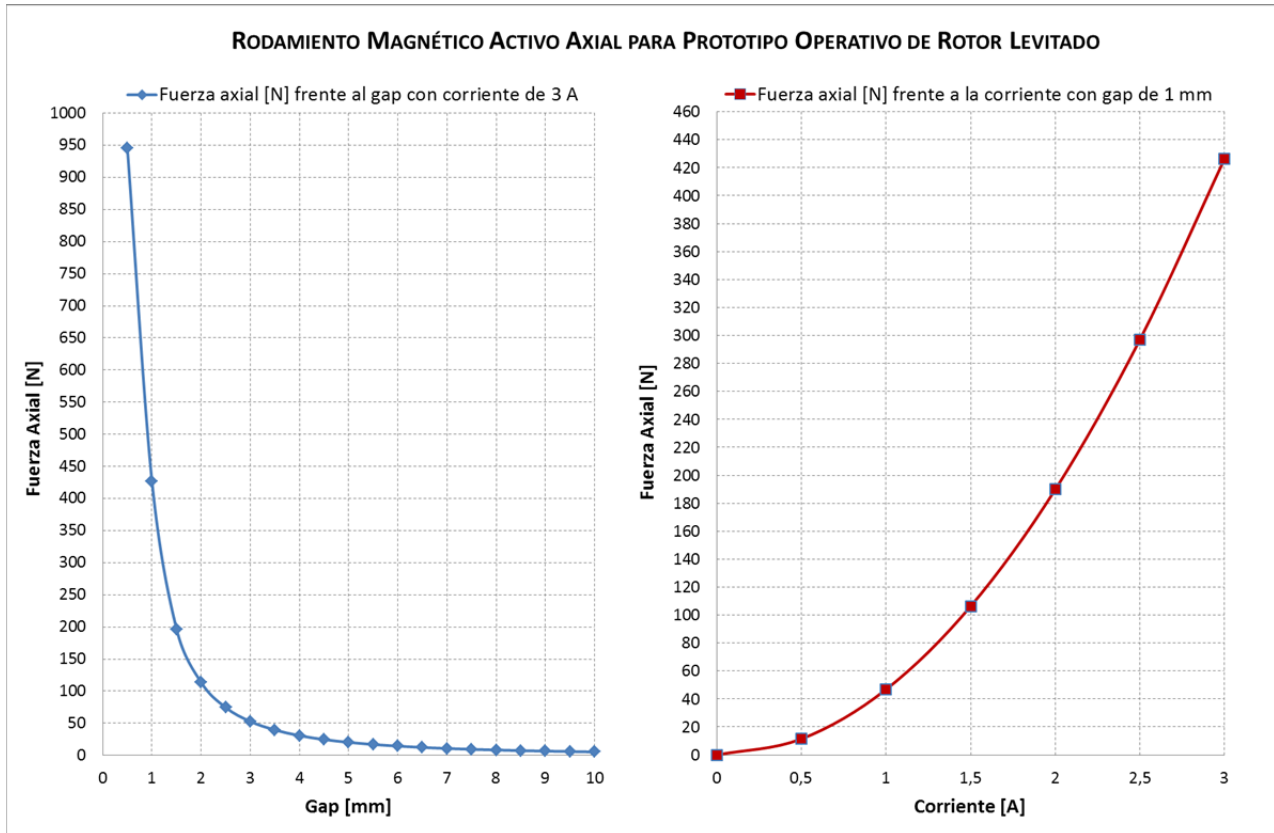
PARÁMETRO	VALOR
Masa del rotor	870 gr
Material del rotor	Acero
Diámetro eje	11-25 mm
Diámetro de la carcasa del estator	250 mm

A partir de estos parámetros y mediante el algoritmo “FAxial\_vs\_Gap.m” diseñado, se va a diseñar un modelo simulado que compense una fuerza axial de 174 N. A dicha fuerza, habrá que añadirle el peso del rotor del modelo del rodamiento magnético activo axial. Tal y como se ha visto en apartados anteriores, mediante modelos de rodamiento magnético activo axial con anillos el doble de anchos que los del modelo publicado en [1], se obtienen elevadas fuerzas a costa de incrementar la masa del disco rotórico. Así, en la Tabla 6.2 se reflejan los parámetros del modelo simulado.

**TABLA 6.2.** PARÁMETROS DEL MODELO DEL RODAMIENTO MAGNÉTICO ACTIVO AXIAL PARA EL PROTOTIPO OPERATIVO DE ROTOR LEVITADO.

PARÁMETRO	VALOR
Corriente DC	3 A
Número de Espiras del Devanado	200 vueltas
Diámetro de Espira	1,95 mm
Radio del Eje	9 mm
Radio del Estator	80 mm
Anchura del Estator	15 mm
Longitud del Estator	50 mm
Radio Interior del Anillo Interno	20 mm
Radio Exterior del Anillo Interno	35 mm
Radio Interior del Anillo Externo	65 mm
Radio Exterior del Anillo Externo	80 mm
Radio del Disco Rotórico	80 mm
Anchura del Disco Rotórico	5 mm
Masa del Disco Rotórico	229,85 gr
Fuerza Axial Total a Compensar	176,3 N

Así, con estos parámetros se han obtenido las gráficas de fuerza axial frente al gap con una corriente constante de 3 A (Figura 6.7 izquierda) y frente a la corriente con el gap del punto de equilibrio de 1 mm (Figura 6.7 derecha).



**Figura 6.7.** Gráfica de fuerza axial frente al gap con una corriente constante de 3 A (izquierda) y frente a la corriente con el gap del punto de equilibrio de 1 mm (derecha) obtenida con FEMM en un rodamiento magnético activo axial diseñado para el prototipo de [8].

En la gráfica de fuerza axial frente al gap, se puede observar que para gaps comprendidos entre 0,5-1,5 mm se obtienen valores de fuerza de 950-200 N respectivamente, con lo que se cubre el valor de fuerza axial estimado en 176,3 N calculado para compensar las fuerzas en dirección axial.

Mientras, en la gráfica de fuerza axial frente a la corriente, se aprecia que para compensar la fuerza hacia abajo en el punto de equilibrio (de unos 60 N) hay que introducir una corriente mínima de unos 1,2 A.

A continuación, se va a calcular el calentamiento de la bobina o pérdidas por efecto Joule. En la Figura 6.8 se refleja el cálculo del devanado realizado con el algoritmo de Matlab.

CÁLCULO DE PARÁMETROS DEL DEVANADO	
Diámetro espira (mm)	1.95
Sección espira (mm <sup>2</sup> )	3
Número de capas	13
Anchura de ventana necesaria (mm)	28.23
Anchura de ventana disponible (mm)	30
¿Diseño correcto?	CORRECTO

**Figura 6.8.** Cálculo del devanado del modelo simulado de rodamiento magnético activo axial para el prototipo de [8].

Así, con estos parámetros se ha obtenido una longitud de la bobina  $l_{bobina}$  y, por consiguiente, una resistencia eléctrica  $R_{bobina}$  de valor:

$$l_{bobina} = 46324,67 \text{ mm} \equiv 46,32 \text{ m}$$

$$R_{bobina} = 0,26 \Omega$$

Por tanto, considerando la corriente máxima  $I$  que circula por la bobina de valor 3 A, se obtienen las siguientes pérdidas por efecto Joule y, por consiguiente, la potencia calórica que genera este devanado:

$$P_{Joule} = I \cdot R_{bobina}^2 = 3 \cdot 0,26^2 = 0,20 \text{ W}$$

## 7. CONCLUSIONES Y TRABAJO FUTURO

En primer lugar, en este TFM se modeliza un rodamiento magnético activo axial basado la topología descrita en [1]. Se realiza también un estudio comparativo entre un modelo simulado de elementos finitos y el modelo validado.

Además, se ha diseñado un algoritmo con Matlab que permite, manteniendo la estructura de la topología descrita en [1], el diseño y análisis del rodamiento magnético activo axial mediante FEMM con los parámetros geométricos que se deseen. Concretamente, se han diseñado dos sub-algoritmos que permiten el cálculo de la fuerza axial ejercida por el estator del rodamiento sobre el rotor frente a la corriente con un gap constante (FAxial\_vs\_Corriente.m) y frente al gap con una corriente constante (FAxial\_vs\_Gap.m), lo que permite un estudio mucho más rápido y dinámico del rodamiento.

Gracias al nuevo algoritmo diseñado, en este TFM se diseña un modelo electromagnético simulado de un rodamiento magnético activo axial que aporta más fuerza axial que el modelo publicado en [1], a costa de un disco rotórico con mayor masa. Así, aumentando la anchura de los anillos electromagnéticos del rodamiento, se ha conseguido un modelo simulado con valores de fuerza axial superiores a los descritos en [1].

Por último, gracias al trabajo previo realizado en este TFM, se ha conseguido un modelo simulado de rodamiento magnético activo axial que compensa las fuerzas en sentido axial del prototipo operativo de rotor levitado descrito en [8].

Como trabajo futuro, quedan pendientes las siguientes tareas de investigación:

- Diseño de un bucle de control con la consigna de corriente adecuada para el control de la estabilidad en dirección axial del rodamiento magnético simulado en este TFM.
- Construcción del modelo de rodamiento magnético activo axial según el modelo simulado en este TFM para acoplarlo al prototipo operativo de rotor levitado de [8].
- Construcción de un segundo rodamiento magnético para mecanizarlo en el extremo opuesto al rodamiento inicial del prototipo operativo de rotor levitado de [8]. Asimismo, diseño del bucle de control con las consignas de corriente adecuadas para el control de la estabilidad en dirección axial de ambos rodamientos magnéticos.



## 8. BIBLIOGRAFÍA

- [1] S. M. Jang, K. H. Kim, K. J. Ko, J. H. Choi, S. Y. Sung and Y. B. Lee, "Characteristic analysis on active thrust magnetic bearings", *2010 International Conference on Electrical Machines and Systems (ICEMS)*, pp. 1710 – 1714, 2010.
- [2] S. L. Chen and C. T. Hsu "Optimal design of a three-pole active magnetic bearing", *IEEE Transactions on Magnetics*, vol. 38, no. 5, pp. 3458 – 3466, 2002.
- [3] A. Hamler, V. Goričan, B. Štumberger, M. Jesenik and M. Trlep "Passive magnetic bearing", *Journal of Magnetism and Magnetic Materials*, vol. 272, pp. 2379 – 2380, 2004.
- [4] A. V. Filatov and E. H. Maslen "Passive magnetic bearing for flywheel energy storage systems", *IEEE Transactions on Magnetics*, vol. 37, no. 6, pp. 3913 – 3924, 2001.
- [5] G. Štumberger, D. Dolinar, U. Pahner and K. Hameyer "Optimization of radial active magnetic bearings using the finite element technique and the differential evolution algorithm", *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 1009 – 1013, 2000.
- [6] M. Autila, E. Lantto and A. Arkkio "Determination of forces and linearized parameters of radial active magnetic bearings by finite element technique", *IEEE Transactions on Magnetics*, vol. 34, no. 3, pp. 684 – 694, 1998.
- [7] S. M. Jang, U. H. Lee and J. Y. Choi "Design and analysis of thrust active magnetic bearing", *Journal of Applied Physics*, vol. 103, pp. 07F122, 2008.
- [8] Proyecto de investigación CYCIT\_2011 de la Universidad de Zaragoza "Rodamientos por levitación magnética para volantes de inercia", ref. ENE2011-29741-C02-02, 2011.
- [9] J. Srinivasa "Introduction to magnetic bearings", *Quality Improvement Program (QIP'08) at Indian Institute of Technology Guwahati*, 2008.
- [10] G. Schweitzer and E. H. Maslen "Magnetic bearings. Theory, design and application to rotating machinery", Capítulos 1 – 2, Ed. Springer, 2009.
- [11] P. K. Agarwala and S. Chand, "Fault tolerant control of three-pole active magnetic bearing", *Expert Systems with Applications*, vol. 36, no. 10, pp. 12592 – 12604, 2009.
- [12] <http://www.mathematicsdictionary.com/>, consultada en Junio de 2012.
- [13] J. G. Bai, X. Z. Zhang and L. M. Wang "A flywheel energy storage system with active magnetic bearings", *Energy Procedia*, vol. 16, pp. 1124 – 1128, 2012.

## 9. ANEXOS

### 9.1 Código fuente del algoritmo de Matlab “FAxial vs Gap.m”

```
function varargout = FAXial_vs_Gap_Matlab(varargin)
% FAXIAL_VS_GAP_MATLAB MATLAB code for FAXial_vs_Gap_Matlab.fig
% FAXIAL_VS_GAP_MATLAB, by itself, creates a new FAXIAL_VS_GAP_MATLAB or raises the existing
% singleton*.
%
% H = FAXIAL_VS_GAP_MATLAB returns the handle to a new FAXIAL_VS_GAP_MATLAB or the handle to
% the existing singleton*.
%
% FAXIAL_VS_GAP_MATLAB('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in FAXIAL_VS_GAP_MATLAB.M with the given input arguments.
%
% FAXIAL_VS_GAP_MATLAB('Property','Value',...) creates a new FAXIAL_VS_GAP_MATLAB or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before FAXial_vs_Gap_Matlab_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to FAXial_vs_Gap_Matlab_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help FAXial_vs_Gap_Matlab

% Last Modified by GUIDE v2.5 09-Sep-2012 12:23:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @FAXial_vs_Gap_Matlab_OpeningFcn, ...
                  'gui_OutputFcn',  @FAXial_vs_Gap_Matlab_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before FAXial_vs_Gap_Matlab is made visible.
function FAXial_vs_Gap_Matlab_OpeningFcn(hObject, eventdata, handles, varargin)
a=imread('Rodamiento_Magnetico.jpg');
image(a);
axis off
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to FAXial_vs_Gap_Matlab (see VARARGIN)

% Choose default command line output for FAXial_vs_Gap_Matlab
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FAXial_vs_Gap_Matlab wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FAXial_vs_Gap_Matlab_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function R_Eje_Callback(hObject, eventdata, handles)
% hObject    handle to R_Eje (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Eje as text
%        str2double(get(hObject,'String')) returns contents of R_Eje as a double

% --- Executes during object creation, after setting all properties.
function R_Eje_CreateFcn(hObject, eventdata, handles)
% hObject    handle to R_Eje (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
set(hObject,'BackgroundColor','white');  
end
```

```
function Intensidad_Callback(hObject, eventdata, handles)  
% hObject    handle to Intensidad (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of Intensidad as text  
%        str2double(get(hObject,'String')) returns contents of Intensidad as a double  
  
% --- Executes during object creation, after setting all properties.  
function Intensidad_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to Intensidad (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
%        See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function Espiras_Callback(hObject, eventdata, handles)  
% hObject    handle to Espiras (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of Espiras as text  
%        str2double(get(hObject,'String')) returns contents of Espiras as a double  
  
% --- Executes during object creation, after setting all properties.  
function Espiras_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to Espiras (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
%        See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function R_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to R_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Estator as text
%         str2double(get(hObject,'String')) returns contents of R_Estator as a double

% --- Executes during object creation, after setting all properties.
function R_Estator_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function A_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to A_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A_Estator as text
%         str2double(get(hObject,'String')) returns contents of A_Estator as a double

% --- Executes during object creation, after setting all properties.
function A_Estator_CreateFcn(hObject, eventdata, handles)
% hObject      handle to A_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function L_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to L_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of L_Estator as text
%         str2double(get(hObject,'String')) returns contents of L_Estator as a double

% --- Executes during object creation, after setting all properties.
function L_Estator_CreateFcn(hObject, eventdata, handles)
% hObject    handle to L_Estator (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rAnilloInterno_Callback(hObject, eventdata, handles)
% hObject    handle to rAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rAnilloInterno as text
%         str2double(get(hObject,'String')) returns contents of rAnilloInterno as a double

% --- Executes during object creation, after setting all properties.
function rAnilloInterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function RadAnilloInterno_Callback(hObject, eventdata, handles)
% hObject    handle to RadAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RadAnilloInterno as text
%         str2double(get(hObject,'String')) returns contents of RadAnilloInterno as a double

% --- Executes during object creation, after setting all properties.
```

```
function RadAnilloInterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RadAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function rAnilloExterno_Callback(hObject, eventdata, handles)
% hObject    handle to rAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rAnilloExterno as text
%         str2double(get(hObject,'String')) returns contents of rAnilloExterno as a double
```

```
% --- Executes during object creation, after setting all properties.
function rAnilloExterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function RadAnilloExterno_Callback(hObject, eventdata, handles)
% hObject    handle to RadAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RadAnilloExterno as text
%         str2double(get(hObject,'String')) returns contents of RadAnilloExterno as a double
```

```
% --- Executes during object creation, after setting all properties.
function RadAnilloExterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RadAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function R_Disco_Callback(hObject, eventdata, handles)
% hObject      handle to R_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Disco as text
%      str2double(get(hObject,'String')) returns contents of R_Disco as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function R_Disco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function A_Disco_Callback(hObject, eventdata, handles)
% hObject      handle to A_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A_Disco as text
%      str2double(get(hObject,'String')) returns contents of A_Disco as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function A_Disco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to A_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```



```

% --- Executes on button press in Boton_Aceptar.
function Boton_Aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to Boton_Aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Corriente = str2num (get(handles.Intensidad,'String'));
Espiras = str2num (get(handles.Espiras,'String'));
R_Eje = str2num (get(handles.R_Eje,'String'));
R_Estator = str2num (get(handles.R_Estator,'String'));
A_Estator = str2num (get(handles.A_Estator,'String'));
LongEstator = str2num (get(handles.L_Estator,'String'));
r_AnilloInterno = str2num (get(handles.r_AnilloInterno,'String'));
R_AnilloInterno = str2num (get(handles.RadAnilloInterno,'String'));
r_AnilloExterno = str2num (get(handles.r_AnilloExterno,'String'));
R_AnilloExterno = str2num (get(handles.RadAnilloExterno,'String'));
R_Disco = str2num (get(handles.R_Disco,'String'));
A_Disco = str2num (get(handles.A_Disco,'String'));

% Comprobamos si hay errores en los parametros
error = comprobacion_errores(Corriente, Espiras, R_Eje, R_Estator, A_Estator, ...
    LongEstator, r_AnilloInterno, R_AnilloInterno, r_AnilloExterno, R_AnilloExterno, ...
    R_Disco, A_Disco);

% Si no hay errores, mostramos un dialogo con los parametros introducidos,
% y preguntamos al usuario si es correcto.
if (error == 0)
    respuesta = questdlg(['I = ',num2str(Corriente),' A'];['Espiras = ',num2str(Espiras),'
vueltas']; ...
        ['R Eje = ',num2str(R_Eje),' mm'];['R Estator = ',num2str(R_Estator),' mm']; ...
        ['A Estator = ',num2str(A_Estator),' mm'];['L Estator = ',num2str(LongEstator),'
mm']; ...
        ['r Anillo interno = ',num2str(r_AnilloInterno),' mm'];['R Anillo interno =
',num2str(R_AnilloInterno),' mm']; ...
        ['r Anillo externo = ',num2str(r_AnilloExterno),' mm'];['R Anillo externo =
',num2str(R_AnilloExterno),' mm']; ...
        ['R Disco = ',num2str(R_Disco),' mm']; ...
        ['A Disco = ',num2str(A_Disco),' mm']; ' '; ' ';}, ...
        'DIMENSIONES', 'Ejecutar FEMM', 'Cancelar', 'Ejecutar FEMM');

% Si hay errores, no ejecutamos FEMM
else
    respuesta = 'Cancelar';
end

% Si no hay errores y el usuario quiere ejecutar femm, entramos aquí
if (strcmp(respuesta, 'Ejecutar FEMM'))

    addpath('c:\\femm42\\mfiles');

```

```
savepath;

openfemm

% Problema tipo Magnetostático -> 0
newdocument(0)

% Frecuencia -> 0 Hz, al trabajar con corriente en DC
% Unidades -> milímetros
% Simetría -> axial o axisimétrica
% El resto de valores por defecto
mi_probdef(0, 'millimeters', 'axi', 1e-8, 1, 30, 0);

% Se cargan los siguientes materiales de la librería de FEMM para el rodamiento
mi_getmaterial('Air');
mi_getmaterial('M-27 Steel');
mi_getmaterial('Copper');

% Se fija el tamaño de la red en 2 mm y el Gap inicial en 1 mm
Gap = 1;
Red = 2;
mi_setgrid(Red, 'cart');

Puntos=[ ];
Puntos(1,1) = r_AnilloInterno-0.75;
Puntos(1,2) = 0;
Puntos(2,1) = R_Estator;
Puntos(2,2) = 0;
Puntos(3,1) = r_AnilloInterno-0.75;
Puntos(3,2) = -A_Estator;
Puntos(4,1) = r_AnilloInterno;
Puntos(4,2) = -A_Estator;
Puntos(5,1) = R_AnilloInterno;
Puntos(5,2) = -A_Estator;
Puntos(6,1) = r_AnilloExterno;
Puntos(6,2) = -A_Estator;
Puntos(7,1) = R_AnilloExterno;
Puntos(7,2) = -A_Estator;
Puntos(8,1) = R_Estator;
Puntos(8,2) = -A_Estator;
Puntos(9,1) = r_AnilloInterno;
Puntos(9,2) = -LongEstator;
Puntos(10,1) = R_AnilloInterno;
Puntos(10,2) = -LongEstator;
Puntos(11,1) = r_AnilloExterno;
Puntos(11,2) = -LongEstator;
Puntos(12,1) = R_AnilloExterno;
Puntos(12,2) = -LongEstator;
Puntos(13,1) = r_AnilloInterno;
Puntos(13,2) = -(LongEstator+Gap);
Puntos(14,1) = R_AnilloInterno;
```

```

Puntos(14,2) = -(LongEstator+Gap);
Puntos(15,1) = r_AnilloExterno;
Puntos(15,2) = -(LongEstator+Gap);
Puntos(16,1) = R_AnilloExterno;
Puntos(16,2) = -(LongEstator+Gap);
Puntos(17,1) = R_Eje;
Puntos(17,2) = -(LongEstator+Gap+1);
Puntos(18,1) = r_AnilloInterno;
Puntos(18,2) = -(LongEstator+Gap+1);
Puntos(19,1) = R_AnilloInterno;
Puntos(19,2) = -(LongEstator+Gap+1);
Puntos(20,1) = r_AnilloExterno;
Puntos(20,2) = -(LongEstator+Gap+1);
Puntos(21,1) = R_AnilloExterno;
Puntos(21,2) = -(LongEstator+Gap+1);
Puntos(22,1) = R_Disco;
Puntos(22,2) = -(LongEstator+Gap+1);
Puntos(23,1) = R_Eje;
Puntos(23,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(24,1) = r_AnilloInterno;
Puntos(24,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(25,1) = R_AnilloInterno;
Puntos(25,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(26,1) = r_AnilloExterno;
Puntos(26,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(27,1) = R_AnilloExterno;
Puntos(27,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(28,1) = R_Disco;
Puntos(28,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(29,1) = r_AnilloInterno;
Puntos(29,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(30,1) = R_AnilloInterno;
Puntos(30,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(31,1) = r_AnilloExterno;
Puntos(31,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(32,1) = R_AnilloExterno;
Puntos(32,2) = -(LongEstator+Gap+2+A_Disco);

% -----Inserción de los puntos con coordenadas (r,z) del rodamiento en FEMM-----
% Inserción de los puntos del estator y del rotor
mi_addnode(Puntos(1,1),Puntos(1,2));
mi_addnode(Puntos(2,1),Puntos(2,2));
mi_addnode(Puntos(3,1),Puntos(3,2));
mi_addnode(Puntos(4,1),Puntos(4,2));
mi_addnode(Puntos(5,1),Puntos(5,2));
mi_addnode(Puntos(6,1),Puntos(6,2));
mi_addnode(Puntos(7,1),Puntos(7,2));
mi_addnode(Puntos(8,1),Puntos(8,2));
mi_addnode(Puntos(9,1),Puntos(9,2));
mi_addnode(Puntos(10,1),Puntos(10,2));
mi_addnode(Puntos(11,1),Puntos(11,2));

```

```

mi_addnode(Puntos(12,1),Puntos(12,2));
mi_addnode(Puntos(13,1),Puntos(13,2));
mi_addnode(Puntos(14,1),Puntos(14,2));
mi_addnode(Puntos(15,1),Puntos(15,2));
mi_addnode(Puntos(16,1),Puntos(16,2));
mi_addnode(Puntos(17,1),Puntos(17,2));
mi_addnode(Puntos(18,1),Puntos(18,2));
mi_addnode(Puntos(19,1),Puntos(19,2));
mi_addnode(Puntos(20,1),Puntos(20,2));
mi_addnode(Puntos(21,1),Puntos(21,2));
mi_addnode(Puntos(22,1),Puntos(22,2));
mi_addnode(Puntos(23,1),Puntos(23,2));
mi_addnode(Puntos(24,1),Puntos(24,2));
mi_addnode(Puntos(25,1),Puntos(25,2));
mi_addnode(Puntos(26,1),Puntos(26,2));
mi_addnode(Puntos(27,1),Puntos(27,2));
mi_addnode(Puntos(28,1),Puntos(28,2));
mi_addnode(Puntos(29,1),Puntos(29,2));
mi_addnode(Puntos(30,1),Puntos(30,2));
mi_addnode(Puntos(31,1),Puntos(31,2));
mi_addnode(Puntos(32,1),Puntos(32,2));

% --Inserción de los puntos de la frontera que engloba al rodamiento.--
% Si el estator es más grande que el disco, el tamaño de la frontera
% se calcula a partir del estator.
if R_Estator >= R_Disco;
    R_Frontera = 2*R_Estator;
else
    R_Frontera = 2*R_Disco;
end
Puntos(33,1) = 0;
Puntos(33,2) = -(R_Frontera + (LongEstator/2));
Puntos(34,1) = 0;
Puntos(34,2) = R_Frontera - (LongEstator/2);
mi_addnode(Puntos(33,1),Puntos(33,2));
mi_addnode(Puntos(34,1),Puntos(34,2));

% -----Inserción de los segmentos o líneas que definen el rodamiento en FEMM-----
% Inserción de los segmentos del estator y del rotor
mi_addsegment(Puntos(1,1),Puntos(1,2),Puntos(2,1),Puntos(2,2));
mi_addsegment(Puntos(1,1),Puntos(1,2),Puntos(3,1),Puntos(3,2));
mi_addsegment(Puntos(2,1),Puntos(2,2),Puntos(8,1),Puntos(8,2));
mi_addsegment(Puntos(3,1),Puntos(3,2),Puntos(4,1),Puntos(4,2));
mi_addsegment(Puntos(4,1),Puntos(4,2),Puntos(9,1),Puntos(9,2));
mi_addsegment(Puntos(5,1),Puntos(5,2),Puntos(6,1),Puntos(6,2));
mi_addsegment(Puntos(5,1),Puntos(5,2),Puntos(10,1),Puntos(10,2));
mi_addsegment(Puntos(6,1),Puntos(6,2),Puntos(11,1),Puntos(11,2));
mi_addsegment(Puntos(7,1),Puntos(7,2),Puntos(8,1),Puntos(8,2));
mi_addsegment(Puntos(7,1),Puntos(7,2),Puntos(12,1),Puntos(12,2));
mi_addsegment(Puntos(9,1),Puntos(9,2),Puntos(10,1),Puntos(10,2));
mi_addsegment(Puntos(10,1),Puntos(10,2),Puntos(11,1),Puntos(11,2));

```

```

mi_addsegment(Puntos(11,1),Puntos(11,2),Puntos(12,1),Puntos(12,2));
mi_addsegment(Puntos(13,1),Puntos(13,2),Puntos(14,1),Puntos(14,2));
mi_addsegment(Puntos(13,1),Puntos(13,2),Puntos(18,1),Puntos(18,2));
mi_addsegment(Puntos(14,1),Puntos(14,2),Puntos(19,1),Puntos(19,2));
mi_addsegment(Puntos(15,1),Puntos(15,2),Puntos(16,1),Puntos(16,2));
mi_addsegment(Puntos(15,1),Puntos(15,2),Puntos(20,1),Puntos(20,2));
mi_addsegment(Puntos(16,1),Puntos(16,2),Puntos(21,1),Puntos(21,2));
mi_addsegment(Puntos(17,1),Puntos(17,2),Puntos(18,1),Puntos(19,2));
mi_addsegment(Puntos(17,1),Puntos(17,2),Puntos(23,1),Puntos(23,2));
mi_addsegment(Puntos(19,1),Puntos(19,2),Puntos(20,1),Puntos(20,2));
mi_addsegment(Puntos(21,1),Puntos(21,2),Puntos(22,1),Puntos(22,2));
mi_addsegment(Puntos(22,1),Puntos(22,2),Puntos(28,1),Puntos(28,2));
mi_addsegment(Puntos(23,1),Puntos(23,2),Puntos(24,1),Puntos(24,2));
mi_addsegment(Puntos(24,1),Puntos(24,2),Puntos(29,1),Puntos(29,2));
mi_addsegment(Puntos(25,1),Puntos(25,2),Puntos(26,1),Puntos(26,2));
mi_addsegment(Puntos(25,1),Puntos(25,2),Puntos(30,1),Puntos(30,2));
mi_addsegment(Puntos(26,1),Puntos(26,2),Puntos(31,1),Puntos(31,2));
mi_addsegment(Puntos(27,1),Puntos(27,2),Puntos(28,1),Puntos(28,2));
mi_addsegment(Puntos(27,1),Puntos(27,2),Puntos(32,1),Puntos(32,2));
mi_addsegment(Puntos(29,1),Puntos(29,2),Puntos(30,1),Puntos(30,2));
mi_addsegment(Puntos(31,1),Puntos(31,2),Puntos(32,1),Puntos(32,2));

% Inserción de los segmentos del entorno que engloba al rodamiento
mi_addsegment(Puntos(33,1),Puntos(33,2),Puntos(34,1),Puntos(34,2));
mi_addarc(Puntos(33,1),Puntos(33,2),Puntos(34,1),Puntos(34,2),180,1);

% -----Se asigna un número de GRUPO a cada parte del rodamiento en FEMM-----
% Grupo 0 -> Devanado estático, Grupo 1 -> Estator, Grupo 2 -> Rotor, Grupo 3 -> Vacío
% El ESTATOR está comprendido por los puntos desde el 1 hasta el punto 12.
for k=1:12;
    mi_selectnode(Puntos(k,1), Puntos(k,2));
end
    mi_setgroup(1);
    mi_clearselected;

% El ROTOR está comprendido por los puntos desde el 13 hasta el punto 32.
for k=13:32;
    mi_selectnode(Puntos(k,1), Puntos(k,2));
end
    mi_setgroup(2);
    mi_clearselected;

% -----Inserción de los materiales del rodamiento en FEMM-----
% Corriente del devanado estático
mi_addcircprop ('Idc', Corriente, 1);

% Material del devanado estático. Grupo -> 0
Mat_Devanado(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Devanado(2)=-(A_Estator+LongEstator)/2;

```

```

mi_addblocklabel(Mat_Devanado(1), Mat_Devanado(2));
mi_selectlabel(Mat_Devanado(1), Mat_Devanado(2));
mi_setblockprop('Copper', 0, 0.25, 'Idc', 0, 0, Espiras);
mi_clearselected;

% Material del estator. Grupo -> 1
Mat_Estator(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Estator(2)=-(A_Estator)/2;
mi_addblocklabel (Mat_Estator(1), Mat_Estator(2));
mi_selectlabel(Mat_Estator(1), Mat_Estator(2));
mi_setblockprop('M-27 Steel', 0, 0.25, ' ', 0, 1, 1);
mi_clearselected;

% Material del rotor. Grupo -> 2
Mat_Rotor(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Rotor(2)=-((LongEstator+Gap+1)+(LongEstator+Gap+1+A_Disco))/2;
mi_addblocklabel(Mat_Rotor(1), Mat_Rotor(2));
mi_selectlabel(Mat_Rotor(1), Mat_Rotor(2));
mi_setblockprop('M-27 Steel', 0, 0.25, ' ', 0, 2, 1);
mi_clearselected;

% Material del vacío. Grupo -> 3
Mat_Vacio(1)=R_Estator+2;
Mat_Vacio(2)=-(A_Estator+LongEstator)/2;
mi_addblocklabel(Mat_Vacio(1), Mat_Vacio(2));
mi_selectlabel(Mat_Vacio(1), Mat_Vacio(2));
mi_setblockprop('Air', 1, 1, ' ', 0, 3, 1);
mi_clearselected;

% -----Condiciones de contorno-----
% Se añade al arco del semicírculo que engloba al rodamiento una condición de contorno para que
el flujo magnético sea tangencial a dicho arco.
% Para ello A0, A1, A2 y phi deben ser cero. El resto de valores se ponen a cero, por defecto.
mi_addboundprop('Tangencial', 0, 0, 0, 0, 0, 0, 0, 0, 0);
u0 = pi*4e-7;
inch = 0.0254;
mi_addboundprop('Espacio infinito', 0, 0, 0, 0, 0, 0, 0, (1/(u0*4*inch)), 0, 2);
mi_selectarcsegment(Puntos(33,1),Puntos(33,2));
mi_setarcsegmentprop(1, 'Tangencial', 0, 0);
mi_clearselected;

% Se hace un zoom del rodamiento magnético en FEMM, no es necesario visualizar el entorno entero
mi_zoom(-5, -(LongEstator+Gap+7+A_Disco), R_Estator*2, 5);

% Se guarda el archivo con el nombre que deseemos sin espacios y con extensión .FEM
mi_saveas('RodMag.FEM');

% ----Se crea el mallado del rodamiento con FEMM----
mi_createmesh;

```

```

% ----Se hace una simulación con FEMM----
mi_analyze(0);
mi_loadsolution;

% ----POSPROCESADOR----
% Se hace zoom de la pieza con FEMM
mo_zoom(-5, -(LongEstator+Gap+7+A_Disco), R_Estator*2, 5);

% Se selecciona el área del rotor (Grupo -> 2) y se calcula la fuerza axial con un bucle "for"
% Suponemos que el Gap varía entre 1-10 mm con intervalos de 0.5 mm, por tanto habrá 19 medidas
n=19;
x=zeros(n,1);
f=zeros(n,1);
traslacion=0.5;

for k=1:n
    mi_analyze;
    mi_loadsolution;
    mo_groupselectblock(2);
    x(k)=Gap+traslacion*(k-1);
    f(k)=mo_blockintegral(19);
    mi_selectgroup(2);
    mi_movetranslate(0, -traslacion);
    mi_clearselected;
end

% Se representa gráficamente la Fuerza axial ejercida en el rotor frente a la distancia de
separación o Gap.
figure;
plot(x,f,'ro');
hold on;
grid on;
plot(x,f,'r-', 'LineWidth', 2);
xlim([0 10]);
xlabel('Gap (mm)');
ylabel('Fuerza axial (N)');
title('Fuerza axial vs. Gap');
save valores_FvsGap.mat x f

closefemm
end

% --- Executes on button press in Boton_Restablecer.
function Boton_Restablecer_Callback(hObject, eventdata, handles)
% hObject    handle to Boton_Restablecer (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
set(handles.Intensidad, 'BackgroundColor', 'red');
set(handles.Espiras, 'BackgroundColor', 'red');
set(handles.R_Eje, 'BackgroundColor', 'red');
set(handles.R_Estator, 'BackgroundColor', 'red');
set(handles.A_Estator, 'BackgroundColor', 'red');
set(handles.L_Estator, 'BackgroundColor', 'red');
set(handles.rAnilloInterno, 'BackgroundColor', 'red');
set(handles.RadAnilloInterno, 'BackgroundColor', 'red');
set(handles.rAnilloExterno, 'BackgroundColor', 'red');
set(handles.RadAnilloExterno, 'BackgroundColor', 'red');
set(handles.R_Disco, 'BackgroundColor', 'red');
set(handles.A_Disco, 'BackgroundColor', 'red');
set(handles.Diametro_espira, 'BackgroundColor', 'red');
set(handles.Seccion_espira, 'BackgroundColor', 'red');
set(handles.Num_capas, 'BackgroundColor', 'red');
set(handles.Aventana_necesaria, 'BackgroundColor', 'red');
set(handles.Aventana_disponible, 'BackgroundColor', 'red');
set(handles.Evaluacion_diseno, 'BackgroundColor', 'red');
pause(1);
set(handles.Intensidad, 'BackgroundColor', 'white');
set(handles.Espiras, 'BackgroundColor', 'white');
set(handles.R_Eje, 'BackgroundColor', 'white');
set(handles.R_Estator, 'BackgroundColor', 'white');
set(handles.A_Estator, 'BackgroundColor', 'white');
set(handles.L_Estator, 'BackgroundColor', 'white');
set(handles.rAnilloInterno, 'BackgroundColor', 'white');
set(handles.RadAnilloInterno, 'BackgroundColor', 'white');
set(handles.rAnilloExterno, 'BackgroundColor', 'white');
set(handles.RadAnilloExterno, 'BackgroundColor', 'white');
set(handles.R_Disco, 'BackgroundColor', 'white');
set(handles.A_Disco, 'BackgroundColor', 'white');
set(handles.Diametro_espira, 'BackgroundColor', 'white');
set(handles.Seccion_espira, 'BackgroundColor', 'white');
set(handles.Num_capas, 'BackgroundColor', 'white');
set(handles.Aventana_necesaria, 'BackgroundColor', 'white');
set(handles.Aventana_disponible, 'BackgroundColor', 'white');
set(handles.Evaluacion_diseno, 'BackgroundColor', 'white');
set(handles.Evaluacion_diseno, 'ForegroundColor', [0.17 0.51 0.34]);

set(handles.Intensidad, 'String', '0.00');
set(handles.Espiras, 'String', '0.00');
set(handles.R_Eje, 'String', '0.00');
set(handles.R_Estator, 'String', '0.00');
set(handles.A_Estator, 'String', '0.00');
set(handles.L_Estator, 'String', '0.00');
set(handles.rAnilloInterno, 'String', '0.00');
set(handles.RadAnilloInterno, 'String', '0.00');
set(handles.rAnilloExterno, 'String', '0.00');
set(handles.RadAnilloExterno, 'String', '0.00');
set(handles.R_Disco, 'String', '0.00');
```



```
set(handles.A_Disco, 'String', '0.00');
set(handles.Diametro_espira, 'String', '0.00');
set(handles.Seccion_espira, 'String', '0.00');
set(handles.Num_capas, 'String', '0.00');
set(handles.Aventana_necesaria, 'String', '0.00');
set(handles.Aventana_disponible, 'String', '0.00');
set(handles.Evaluacion_diseno, 'String', '-');

function Gap_Callback(hObject, eventdata, handles)
% hObject    handle to Gap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Gap as text
%        str2double(get(hObject,'String')) returns contents of Gap as a double

% --- Executes during object creation, after setting all properties.
function Gap_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Gap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Diametro_espira_Callback(hObject, eventdata, handles)
% hObject    handle to Diametro_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Diametro_espira as text
%        str2double(get(hObject,'String')) returns contents of Diametro_espira as a double

% --- Executes during object creation, after setting all properties.
function Diametro_espira_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Diametro_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Seccion_espira_Callback(hObject, eventdata, handles)
% hObject      handle to Seccion_espira (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Seccion_espira as text
%      str2double(get(hObject,'String')) returns contents of Seccion_espira as a double

% --- Executes during object creation, after setting all properties.
function Seccion_espira_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Seccion_espira (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Aventana_necesaria_Callback(hObject, eventdata, handles)
% hObject      handle to Aventana_necesaria (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aventana_necesaria as text
%      str2double(get(hObject,'String')) returns contents of Aventana_necesaria as a double

% --- Executes during object creation, after setting all properties.
function Aventana_necesaria_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Aventana_necesaria (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

function edit18_Callback(hObject, eventdata, handles)
% hObject      handle to edit18 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%         str2double(get(hObject,'String')) returns contents of edit18 as a double

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit18 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Boton_CalculoDevanado.
function Boton_CalculoDevanado_Callback(hObject, eventdata, handles)
% hObject      handle to Boton_CalculoDevanado (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
Corriente = str2num (get(handles.Intensidad,'String'));
Espiras = str2num (get(handles.Espiras,'String'));
R_Eje = str2num (get(handles.R_Eje,'String'));
R_Estator = str2num (get(handles.R_Estator,'String'));
A_Estator = str2num (get(handles.A_Estator,'String'));
LongEstator = str2num (get(handles.L_Estator,'String'));
r_AnilloInterno = str2num (get(handles.rAnilloInterno,'String'));
R_AnilloInterno = str2num (get(handles.RadAnilloInterno,'String'));
r_AnilloExterno = str2num (get(handles.rAnilloExterno,'String'));
R_AnilloExterno = str2num (get(handles.RadAnilloExterno,'String'));
R_Disco = str2num (get(handles.R_Disco,'String'));
A_Disco = str2num (get(handles.A_Disco,'String'));

L_Ventana = LongEstator - A_Estator;
Densidad_max = 1;
Factor_empaquetado = 0.9;
S_Espira = Corriente/Densidad_max;
R_Espira = sqrt(S_Espira/pi);
D_Espira = 2*R_Espira;
Espiras_Capa = floor((L_Ventana*Factor_empaquetado)/D_Espira);
Capas = ceil(Espiras/Espiras_Capa);
A_ventana_necesaria = (Capas*D_Espira)/Factor_empaquetado;

```

```

A_ventana_disponible = r_AnilloExterno - R_AnilloInterno;
if A_ventana_disponible >= A_ventana_necesaria
    set(handles.Evaluacion_diseno, 'ForegroundColor', 'green');
    set(handles.Evaluacion_diseno, 'String', 'CORRECTO');
else
    set(handles.Evaluacion_diseno, 'ForegroundColor', 'red');
    set(handles.Evaluacion_diseno, 'String', 'INCORRECTO');
end

set(handles.Diametro_espira, 'String', num2str(roundn(D_Espira,-2)));
set(handles.Seccion_espira, 'String', num2str(roundn(S_Espira,-2)));
set(handles.Num_capas, 'String', num2str(roundn(Capas,0)));
set(handles.Aventana_necesaria, 'String', num2str(roundn(A_ventana_necesaria,-2)));
set(handles.Aventana_disponible, 'String', num2str(roundn(A_ventana_disponible,-2)));

function Aventana_disponible_Callback(hObject, eventdata, handles)
% hObject    handle to Aventana_disponible (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aventana_disponible as text
%        str2double(get(hObject,'String')) returns contents of Aventana_disponible as a double

% --- Executes during object creation, after setting all properties.
function Aventana_disponible_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aventana_disponible (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Num_capas_Callback(hObject, eventdata, handles)
% hObject    handle to Num_capas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Num_capas as text
%        str2double(get(hObject,'String')) returns contents of Num_capas as a double

% --- Executes during object creation, after setting all properties.

```

```
function Num_capas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Num_capas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Evaluacion_diseno_Callback(hObject, eventdata, handles)
% hObject    handle to Evaluacion_diseno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Evaluacion_diseno as text
%         str2double(get(hObject,'String')) returns contents of Evaluacion_diseno as a double

% --- Executes during object creation, after setting all properties.
function Evaluacion_diseno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Evaluacion_diseno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

## 9.2 Código fuente del algoritmo de Matlab “FAXial vs Corriente.m”

```
function varargout = FAXial_vs_Corriente_Matlab(varargin)
% FAXIAL_VS_CORRIENTE_MATLAB MATLAB code for FAXial_vs_Corriente_Matlab.fig
%     FAXIAL_VS_CORRIENTE_MATLAB, by itself, creates a new FAXIAL_VS_CORRIENTE_MATLAB or raises the
existing
%     singleton*.
%
%     H = FAXIAL_VS_CORRIENTE_MATLAB returns the handle to a new FAXIAL_VS_CORRIENTE_MATLAB or the
handle to
%     the existing singleton*.
%
%     FAXIAL_VS_CORRIENTE_MATLAB('CALLBACK',hObject,eventData,handles,...) calls the local
function named CALLBACK in FAXIAL_VS_CORRIENTE_MATLAB.M with the given input arguments.
%
%     FAXIAL_VS_CORRIENTE_MATLAB('Property','Value',...) creates a new FAXIAL_VS_CORRIENTE_MATLAB or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before FAXial_vs_Corriente_Matlab_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to FAXial_vs_Corriente_Matlab_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help FAXial_vs_Corriente_Matlab

% Last Modified by GUIDE v2.5 09-Sep-2012 13:05:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @FAXial_vs_Corriente_Matlab_OpeningFcn, ...
                  'gui_OutputFcn',  @FAXial_vs_Corriente_Matlab_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before FAXial_vs_Corriente_Matlab is made visible.
function FAXial_vs_Corriente_Matlab_OpeningFcn(hObject, eventdata, handles, varargin)
a=imread('Rodamiento_Magnetico.jpg');
image(a);
axis off
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to FAXial_vs_Corriente_Matlab (see VARARGIN)

% Choose default command line output for FAXial_vs_Corriente_Matlab
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FAXial_vs_Corriente_Matlab wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FAXial_vs_Corriente_Matlab_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function R_Eje_Callback(hObject, eventdata, handles)
% hObject    handle to R_Eje (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Eje as text
%        str2double(get(hObject,'String')) returns contents of R_Eje as a double

% --- Executes during object creation, after setting all properties.
function R_Eje_CreateFcn(hObject, eventdata, handles)
% hObject    handle to R_Eje (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
set(hObject,'BackgroundColor','white');  
end
```

```
function Intensidad_Callback(hObject, eventdata, handles)  
% hObject    handle to Intensidad (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of Intensidad as text  
%        str2double(get(hObject,'String')) returns contents of Intensidad as a double  
  
% --- Executes during object creation, after setting all properties.  
function Intensidad_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to Intensidad (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function Espiras_Callback(hObject, eventdata, handles)  
% hObject    handle to Espiras (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of Espiras as text  
%        str2double(get(hObject,'String')) returns contents of Espiras as a double  
  
% --- Executes during object creation, after setting all properties.  
function Espiras_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to Espiras (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```



```
function R_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to R_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Estator as text
%         str2double(get(hObject,'String')) returns contents of R_Estator as a double

% --- Executes during object creation, after setting all properties.
function R_Estator_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function A_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to A_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A_Estator as text
%         str2double(get(hObject,'String')) returns contents of A_Estator as a double

% --- Executes during object creation, after setting all properties.
function A_Estator_CreateFcn(hObject, eventdata, handles)
% hObject      handle to A_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function L_Estator_Callback(hObject, eventdata, handles)
% hObject      handle to L_Estator (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of L_Estator as text
%         str2double(get(hObject,'String')) returns contents of L_Estator as a double

% --- Executes during object creation, after setting all properties.
function L_Estator_CreateFcn(hObject, eventdata, handles)
% hObject    handle to L_Estator (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rAnilloInterno_Callback(hObject, eventdata, handles)
% hObject    handle to rAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rAnilloInterno as text
%         str2double(get(hObject,'String')) returns contents of rAnilloInterno as a double

% --- Executes during object creation, after setting all properties.
function rAnilloInterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function RadAnilloInterno_Callback(hObject, eventdata, handles)
% hObject    handle to RadAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RadAnilloInterno as text
%         str2double(get(hObject,'String')) returns contents of RadAnilloInterno as a double

% --- Executes during object creation, after setting all properties.
```

```
function RadAnilloInterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RadAnilloInterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function rAnilloExterno_Callback(hObject, eventdata, handles)
% hObject    handle to rAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rAnilloExterno as text
%         str2double(get(hObject,'String')) returns contents of rAnilloExterno as a double
```

```
% --- Executes during object creation, after setting all properties.
function rAnilloExterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function RadAnilloExterno_Callback(hObject, eventdata, handles)
% hObject    handle to RadAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RadAnilloExterno as text
%         str2double(get(hObject,'String')) returns contents of RadAnilloExterno as a double
```

```
% --- Executes during object creation, after setting all properties.
function RadAnilloExterno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RadAnilloExterno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function R_Disco_Callback(hObject, eventdata, handles)
% hObject      handle to R_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of R_Disco as text
%      str2double(get(hObject,'String')) returns contents of R_Disco as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function R_Disco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function A_Disco_Callback(hObject, eventdata, handles)
% hObject      handle to A_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A_Disco as text
%      str2double(get(hObject,'String')) returns contents of A_Disco as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function A_Disco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to A_Disco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

% --- Executes on button press in Boton_Aceptar.
function Boton_Aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to Boton_Aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Espiras = str2num (get(handles.Espiras,'String'));
R_Eje = str2num (get(handles.R_Eje,'String'));
R_Estator = str2num (get(handles.R_Estator,'String'));
A_Estator = str2num (get(handles.A_Estator,'String'));
LongEstator = str2num (get(handles.L_Estator,'String'));
r_AnilloInterno = str2num (get(handles.r_AnilloInterno,'String'));
R_AnilloInterno = str2num (get(handles.RadAnilloInterno,'String'));
r_AnilloExterno = str2num (get(handles.r_AnilloExterno,'String'));
R_AnilloExterno = str2num (get(handles.RadAnilloExterno,'String'));
Gap = str2num (get(handles.Gap,'String'));
R_Disco = str2num (get(handles.R_Disco,'String'));
A_Disco = str2num (get(handles.A_Disco,'String'));

% Comprobamos si hay errores en los parametros
error = comprobacion_errores2(Espiras, R_Eje, R_Estator, A_Estator, ...
    LongEstator, r_AnilloInterno, R_AnilloInterno, r_AnilloExterno, R_AnilloExterno, ...
    Gap, R_Disco, A_Disco);

% Si no hay errores, mostramos un dialogo con los parametros introducidos,
% y preguntamos al usuario si es correcto.
if (error == 0)
    respuesta = questdlg(['Espiras = ',num2str(Espiras),' vueltas']; ...
        ['R Eje = ',num2str(R_Eje),' mm'];['R Estator = ',num2str(R_Estator),' mm']; ...
        ['A Estator = ',num2str(A_Estator),' mm'];['L Estator = ',num2str(LongEstator),'
mm']; ...
        ['r Anillo interno = ',num2str(r_AnilloInterno),' mm'];['R Anillo interno =
',num2str(R_AnilloInterno),' mm']; ...
        ['r Anillo externo = ',num2str(r_AnilloExterno),' mm'];['R Anillo externo =
',num2str(R_AnilloExterno),' mm']; ...
        ['Gap = ',num2str(Gap),' mm'];['R Disco = ',num2str(R_Disco),' mm']; ...
        ['A Disco = ',num2str(A_Disco),' mm']; ' '; ' '];, ...
        'DIMENSIONES', 'Ejecutar FEMM', 'Cancelar', 'Ejecutar FEMM');

% Si hay errores, no ejecutamos FEMM
else
    respuesta = 'Cancelar';
end

% Si no hay errores y el usuario quiere ejecutar femm, entramos aquí
if (strcmp(respuesta, 'Ejecutar FEMM'))

    addpath('c:\femm42\mfiles');
    savepath;

```

```
openfemm

% Problema tipo Magnetostático -> 0
newdocument(0)

% Frecuencia -> 0 Hz, al trabajar con corriente en DC
% Unidades -> milímetros
% Simetría -> axial o axisimétrica
% El resto de valores por defecto
mi_probdef(0, 'millimeters', 'axi', 1e-8, 1, 30, 0);

% Se cargan los siguientes materiales de la librería de FEMM para el rodamiento
mi_getmaterial('Air');
mi_getmaterial('M-27 Steel');
mi_getmaterial('Copper');

% Se fija el tamaño de la red en 2 mm
Red = 2;
mi_setgrid(Red, 'cart');

Puntos=[ ];
Puntos(1,1) = r_AnilloInterno-0.75;
Puntos(1,2) = 0;
Puntos(2,1) = R_Estator;
Puntos(2,2) = 0;
Puntos(3,1) = r_AnilloInterno-0.75;
Puntos(3,2) = -A_Estator;
Puntos(4,1) = r_AnilloInterno;
Puntos(4,2) = -A_Estator;
Puntos(5,1) = R_AnilloInterno;
Puntos(5,2) = -A_Estator;
Puntos(6,1) = r_AnilloExterno;
Puntos(6,2) = -A_Estator;
Puntos(7,1) = R_AnilloExterno;
Puntos(7,2) = -A_Estator;
Puntos(8,1) = R_Estator;
Puntos(8,2) = -A_Estator;
Puntos(9,1) = r_AnilloInterno;
Puntos(9,2) = -LongEstator;
Puntos(10,1) = R_AnilloInterno;
Puntos(10,2) = -LongEstator;
Puntos(11,1) = r_AnilloExterno;
Puntos(11,2) = -LongEstator;
Puntos(12,1) = R_AnilloExterno;
Puntos(12,2) = -LongEstator;
Puntos(13,1) = r_AnilloInterno;
Puntos(13,2) = -(LongEstator+Gap);
Puntos(14,1) = R_AnilloInterno;
Puntos(14,2) = -(LongEstator+Gap);
Puntos(15,1) = r_AnilloExterno;
```

```

Puntos(15,2) = -(LongEstator+Gap);
Puntos(16,1) = R_AnilloExterno;
Puntos(16,2) = -(LongEstator+Gap);
Puntos(17,1) = R_Eje;
Puntos(17,2) = -(LongEstator+Gap+1);
Puntos(18,1) = r_AnilloInterno;
Puntos(18,2) = -(LongEstator+Gap+1);
Puntos(19,1) = R_AnilloInterno;
Puntos(19,2) = -(LongEstator+Gap+1);
Puntos(20,1) = r_AnilloExterno;
Puntos(20,2) = -(LongEstator+Gap+1);
Puntos(21,1) = R_AnilloExterno;
Puntos(21,2) = -(LongEstator+Gap+1);
Puntos(22,1) = R_Disco;
Puntos(22,2) = -(LongEstator+Gap+1);
Puntos(23,1) = R_Eje;
Puntos(23,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(24,1) = r_AnilloInterno;
Puntos(24,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(25,1) = R_AnilloInterno;
Puntos(25,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(26,1) = r_AnilloExterno;
Puntos(26,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(27,1) = R_AnilloExterno;
Puntos(27,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(28,1) = R_Disco;
Puntos(28,2) = -(LongEstator+Gap+1+A_Disco);
Puntos(29,1) = r_AnilloInterno;
Puntos(29,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(30,1) = R_AnilloInterno;
Puntos(30,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(31,1) = r_AnilloExterno;
Puntos(31,2) = -(LongEstator+Gap+2+A_Disco);
Puntos(32,1) = R_AnilloExterno;
Puntos(32,2) = -(LongEstator+Gap+2+A_Disco);

% -----Inserción de los puntos con coordenadas (r,z) del rodamiento en FEMM-----
% Inserción de los puntos del estator y del rotor
mi_addnode(Puntos(1,1),Puntos(1,2));
mi_addnode(Puntos(2,1),Puntos(2,2));
mi_addnode(Puntos(3,1),Puntos(3,2));
mi_addnode(Puntos(4,1),Puntos(4,2));
mi_addnode(Puntos(5,1),Puntos(5,2));
mi_addnode(Puntos(6,1),Puntos(6,2));
mi_addnode(Puntos(7,1),Puntos(7,2));
mi_addnode(Puntos(8,1),Puntos(8,2));
mi_addnode(Puntos(9,1),Puntos(9,2));
mi_addnode(Puntos(10,1),Puntos(10,2));
mi_addnode(Puntos(11,1),Puntos(11,2));
mi_addnode(Puntos(12,1),Puntos(12,2));
mi_addnode(Puntos(13,1),Puntos(13,2));

```

```

mi_addnode(Puntos(14,1),Puntos(14,2));
mi_addnode(Puntos(15,1),Puntos(15,2));
mi_addnode(Puntos(16,1),Puntos(16,2));
mi_addnode(Puntos(17,1),Puntos(17,2));
mi_addnode(Puntos(18,1),Puntos(18,2));
mi_addnode(Puntos(19,1),Puntos(19,2));
mi_addnode(Puntos(20,1),Puntos(20,2));
mi_addnode(Puntos(21,1),Puntos(21,2));
mi_addnode(Puntos(22,1),Puntos(22,2));
mi_addnode(Puntos(23,1),Puntos(23,2));
mi_addnode(Puntos(24,1),Puntos(24,2));
mi_addnode(Puntos(25,1),Puntos(25,2));
mi_addnode(Puntos(26,1),Puntos(26,2));
mi_addnode(Puntos(27,1),Puntos(27,2));
mi_addnode(Puntos(28,1),Puntos(28,2));
mi_addnode(Puntos(29,1),Puntos(29,2));
mi_addnode(Puntos(30,1),Puntos(30,2));
mi_addnode(Puntos(31,1),Puntos(31,2));
mi_addnode(Puntos(32,1),Puntos(32,2));

% Inserción de los puntos de la frontera que engloba al rodamiento
R_Frontera = 2*R_Estator;
Puntos(33,1) = 0;
Puntos(33,2) = -(R_Frontera + (LongEstator/2));
Puntos(34,1) = 0;
Puntos(34,2) = R_Frontera - (LongEstator/2);
mi_addnode(Puntos(33,1),Puntos(33,2));
mi_addnode(Puntos(34,1),Puntos(34,2));

% -----Inserción de los segmentos o líneas que definen el rodamiento en FEMM-----
% Inserción de los segmentos del estator y del rotor
mi_addsegment(Puntos(1,1),Puntos(1,2),Puntos(2,1),Puntos(2,2));
mi_addsegment(Puntos(1,1),Puntos(1,2),Puntos(3,1),Puntos(3,2));
mi_addsegment(Puntos(2,1),Puntos(2,2),Puntos(8,1),Puntos(8,2));
mi_addsegment(Puntos(3,1),Puntos(3,2),Puntos(4,1),Puntos(4,2));
mi_addsegment(Puntos(4,1),Puntos(4,2),Puntos(9,1),Puntos(9,2));
mi_addsegment(Puntos(5,1),Puntos(5,2),Puntos(6,1),Puntos(6,2));
mi_addsegment(Puntos(5,1),Puntos(5,2),Puntos(10,1),Puntos(10,2));
mi_addsegment(Puntos(6,1),Puntos(6,2),Puntos(11,1),Puntos(11,2));
mi_addsegment(Puntos(7,1),Puntos(7,2),Puntos(8,1),Puntos(8,2));
mi_addsegment(Puntos(7,1),Puntos(7,2),Puntos(12,1),Puntos(12,2));
mi_addsegment(Puntos(9,1),Puntos(9,2),Puntos(10,1),Puntos(10,2));
mi_addsegment(Puntos(10,1),Puntos(10,2),Puntos(11,1),Puntos(11,2));
mi_addsegment(Puntos(11,1),Puntos(11,2),Puntos(12,1),Puntos(12,2));
mi_addsegment(Puntos(13,1),Puntos(13,2),Puntos(14,1),Puntos(14,2));
mi_addsegment(Puntos(13,1),Puntos(13,2),Puntos(18,1),Puntos(18,2));
mi_addsegment(Puntos(14,1),Puntos(14,2),Puntos(19,1),Puntos(19,2));
mi_addsegment(Puntos(15,1),Puntos(15,2),Puntos(16,1),Puntos(16,2));
mi_addsegment(Puntos(15,1),Puntos(15,2),Puntos(20,1),Puntos(20,2));
mi_addsegment(Puntos(16,1),Puntos(16,2),Puntos(21,1),Puntos(21,2));

```



```

mi_addsegment(Puntos(17,1),Puntos(17,2),Puntos(18,1),Puntos(19,2));
mi_addsegment(Puntos(17,1),Puntos(17,2),Puntos(23,1),Puntos(23,2));
mi_addsegment(Puntos(19,1),Puntos(19,2),Puntos(20,1),Puntos(20,2));
mi_addsegment(Puntos(21,1),Puntos(21,2),Puntos(22,1),Puntos(22,2));
mi_addsegment(Puntos(22,1),Puntos(22,2),Puntos(28,1),Puntos(28,2));
mi_addsegment(Puntos(23,1),Puntos(23,2),Puntos(24,1),Puntos(24,2));
mi_addsegment(Puntos(24,1),Puntos(24,2),Puntos(29,1),Puntos(29,2));
mi_addsegment(Puntos(25,1),Puntos(25,2),Puntos(26,1),Puntos(26,2));
mi_addsegment(Puntos(25,1),Puntos(25,2),Puntos(30,1),Puntos(30,2));
mi_addsegment(Puntos(26,1),Puntos(26,2),Puntos(31,1),Puntos(31,2));
mi_addsegment(Puntos(27,1),Puntos(27,2),Puntos(28,1),Puntos(28,2));
mi_addsegment(Puntos(27,1),Puntos(27,2),Puntos(32,1),Puntos(32,2));
mi_addsegment(Puntos(29,1),Puntos(29,2),Puntos(30,1),Puntos(30,2));
mi_addsegment(Puntos(31,1),Puntos(31,2),Puntos(32,1),Puntos(32,2));

% Inserción de los segmentos del entorno que engloba al rodamiento
mi_addsegment(Puntos(33,1),Puntos(33,2),Puntos(34,1),Puntos(34,2));
mi_addarc(Puntos(33,1),Puntos(33,2),Puntos(34,1),Puntos(34,2),180,1);

% -----Se asigna un número de GRUPO a cada parte del rodamiento en FEMM-----
% Grupo 0 -> Devanado estático, Grupo 1 -> Estator, Grupo 2 -> Rotor, Grupo 3 -> Vacío
% El ESTATOR está comprendido por los puntos desde el 1 hasta el punto 12.
for k=1:12;
    mi_selectnode(Puntos(k,1), Puntos(k,2));
end
    mi_setgroup(1);
    mi_clearselected;

% El ROTOR está comprendido por los puntos desde el 13 hasta el punto 32.
for k=13:32;
    mi_selectnode(Puntos(k,1), Puntos(k,2));
end
    mi_setgroup(2);
    mi_clearselected;

% -----Inserción de los materiales del rodamiento en FEMM-----
% Corriente del devanado estático. Se establece una corriente inicial de 0A.
Corriente = 0;
mi_addcircprop ('Idc', Corriente, 1);

% Material del devanado estático. Grupo -> 0
Mat_Devanado(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Devanado(2)=-(A_Estator+LongEstator)/2;
mi_addblocklabel(Mat_Devanado(1), Mat_Devanado(2));
mi_selectlabel(Mat_Devanado(1), Mat_Devanado(2));
mi_setblockprop('Copper', 0, 0.25, 'Idc', 0 , 0, Espiras);
mi_clearselected;

% Material del estator. Grupo -> 1

```

```
Mat_Estator(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Estator(2)=-(A_Estator)/2;
mi_addblocklabel (Mat_Estator(1), Mat_Estator(2));
mi_selectlabel(Mat_Estator(1), Mat_Estator(2));
mi_setblockprop('M-27 Steel', 0, 0.25, ' ', 0, 1, 1);
mi_clearselected;

% Material del rotor. Grupo -> 2
Mat_Rotor(1)=(R_AnilloInterno+r_AnilloExterno)/2;
Mat_Rotor(2)=-((LongEstator+Gap+1)+(LongEstator+Gap+1+A_Disco))/2;
mi_addblocklabel(Mat_Rotor(1), Mat_Rotor(2));
mi_selectlabel(Mat_Rotor(1), Mat_Rotor(2));
mi_setblockprop('M-27 Steel', 0, 0.25, ' ', 0, 2, 1);
mi_clearselected;

% Material del vacío. Grupo -> 3
Mat_Vacio(1)=R_Estator+2;
Mat_Vacio(2)=-(A_Estator+LongEstator)/2;
mi_addblocklabel(Mat_Vacio(1), Mat_Vacio(2));
mi_selectlabel(Mat_Vacio(1), Mat_Vacio(2));
mi_setblockprop('Air', 1, 1, ' ', 0, 3, 1);
mi_clearselected;

% -----Condiciones de contorno-----
% Se añade al arco del semicírculo que engloba al rodamiento una condición de contorno para que
el flujo magnético sea tangencial a dicho arco.
% Para ello A0, A1, A2 y phi deben ser cero. El resto de valores se ponen a cero, por defecto.
mi_addboundprop('Tangencial', 0, 0, 0, 0, 0, 0, 0, 0, 0);
mi_selectarcsegment(Puntos(33,1),Puntos(33,2));
mi_setarcsegmentprop(1, 'Tangencial', 0, 0);
mi_clearselected;

% Se hace un zoom del rodamiento magnético en FEMM, no es necesario visualizar el entorno entero
mi_zoom(-5, -(LongEstator+Gap+7+A_Disco), R_Estator*2, 5);

% Se guarda el archivo con el nombre que deseemos sin espacios y con extensión .FEM
mi_saveas('RodMag.FEM');

% ----Se crea el mallado del rodamiento con FEMM----
mi_createmesh;

% ----Se hace una simulación con FEMM----
mi_analyze(0);
mi_loadsolution;

% ----POSPROCESADOR----
% Se hace zoom de la pieza con FEMM
mo_zoom(-5, -(LongEstator+Gap+7+A_Disco), R_Estator*2, 5);
```

```

% Se selecciona el área del rotor (Grupo -> 2) y se calcula la fuerza axial con un bucle "for"
% Suponemos que la corriente varía entre 0-3 A con intervalos de 0.5 A, por tanto habrá 7 medidas
n=7;
x=zeros(n,1);
f=zeros(n,1);
incremento_I=0.5;

for k=1:n
    mi_analyze;
    mi_loadsolution;
    mo_groupselectblock(2);
    x(k)=incremento_I*(k-1);
    f(k)=mo_blockintegral(19);
    Corriente=incremento_I*k;
    mi_setcurrent('Idc', Corriente);
end

% Se representa gráficamente la Fuerza axial ejercida en el rotor frente a la Corriente del
devanado estático.
figure;
plot(x,f,'bo');
hold on;
grid on;
plot(x,f,'b-', 'LineWidth', 2);
xlim([0 3]);
xlabel('Corriente (A)');
ylabel('Fuerza axial (N)');
title('Fuerza axial vs. Corriente');
save valores_FvsI.mat x f

closefemm
end

% --- Executes on button press in Boton_Restablecer.
function Boton_Restablecer_Callback(hObject, eventdata, handles)
% hObject    handle to Boton_Restablecer (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Espiras, 'BackgroundColor', 'red');
set(handles.R_Eje, 'BackgroundColor', 'red');
set(handles.R_Estator, 'BackgroundColor', 'red');
set(handles.A_Estator, 'BackgroundColor', 'red');
set(handles.L_Estator, 'BackgroundColor', 'red');
set(handles.rAnilloInterno, 'BackgroundColor', 'red');
set(handles.RadAnilloInterno, 'BackgroundColor', 'red');
set(handles.rAnilloExterno, 'BackgroundColor', 'red');
set(handles.RadAnilloExterno, 'BackgroundColor', 'red');
set(handles.Gap, 'BackgroundColor', 'red');

```

```
set(handles.R_Disco, 'BackgroundColor', 'red');
set(handles.A_Disco, 'BackgroundColor', 'red');
set(handles.Diametro_espira, 'BackgroundColor', 'red');
set(handles.Seccion_espira, 'BackgroundColor', 'red');
set(handles.Num_capas, 'BackgroundColor', 'red');
set(handles.Aventana_necesaria, 'BackgroundColor', 'red');
set(handles.Aventana_disponible, 'BackgroundColor', 'red');
set(handles.Evaluacion_diseno, 'BackgroundColor', 'red');
pause(1);
set(handles.Espiras, 'BackgroundColor', 'white');
set(handles.R_Eje, 'BackgroundColor', 'white');
set(handles.R_Estator, 'BackgroundColor', 'white');
set(handles.A_Estator, 'BackgroundColor', 'white');
set(handles.L_Estator, 'BackgroundColor', 'white');
set(handles.rAnilloInterno, 'BackgroundColor', 'white');
set(handles.RadAnilloInterno, 'BackgroundColor', 'white');
set(handles.rAnilloExterno, 'BackgroundColor', 'white');
set(handles.RadAnilloExterno, 'BackgroundColor', 'white');
set(handles.Gap, 'BackgroundColor', 'white');
set(handles.R_Disco, 'BackgroundColor', 'white');
set(handles.A_Disco, 'BackgroundColor', 'white');
set(handles.Diametro_espira, 'BackgroundColor', 'white');
set(handles.Seccion_espira, 'BackgroundColor', 'white');
set(handles.Num_capas, 'BackgroundColor', 'white');
set(handles.Aventana_necesaria, 'BackgroundColor', 'white');
set(handles.Aventana_disponible, 'BackgroundColor', 'white');
set(handles.Evaluacion_diseno, 'BackgroundColor', 'white');
set(handles.Evaluacion_diseno, 'ForegroundColor', [0.17 0.51 0.34]);

set(handles.Espiras, 'String', '0.00');
set(handles.R_Eje, 'String', '0.00');
set(handles.R_Estator, 'String', '0.00');
set(handles.A_Estator, 'String', '0.00');
set(handles.L_Estator, 'String', '0.00');
set(handles.rAnilloInterno, 'String', '0.00');
set(handles.RadAnilloInterno, 'String', '0.00');
set(handles.rAnilloExterno, 'String', '0.00');
set(handles.RadAnilloExterno, 'String', '0.00');
set(handles.Gap, 'String', '0.00');
set(handles.R_Disco, 'String', '0.00');
set(handles.A_Disco, 'String', '0.00');
set(handles.Diametro_espira, 'String', '0.00');
set(handles.Seccion_espira, 'String', '0.00');
set(handles.Num_capas, 'String', '0.00');
set(handles.Aventana_necesaria, 'String', '0.00');
set(handles.Aventana_disponible, 'String', '0.00');
set(handles.Evaluacion_diseno, 'String', '-');
```

```
function Gap_Callback(hObject, eventdata, handles)
% hObject    handle to Gap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Gap as text
%        str2double(get(hObject,'String')) returns contents of Gap as a double

% --- Executes during object creation, after setting all properties.
function Gap_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Gap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Diametro_espira_Callback(hObject, eventdata, handles)
% hObject    handle to Diametro_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Diametro_espira as text
%        str2double(get(hObject,'String')) returns contents of Diametro_espira as a double

% --- Executes during object creation, after setting all properties.
function Diametro_espira_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Diametro_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Seccion_espira_Callback(hObject, eventdata, handles)
% hObject    handle to Seccion_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Seccion_espira as text
%         str2double(get(hObject,'String')) returns contents of Seccion_espira as a double


% --- Executes during object creation, after setting all properties.
function Seccion_espira_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Seccion_espira (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function Aventana_necesaria_Callback(hObject, eventdata, handles)
% hObject    handle to Aventana_necesaria (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of Aventana_necesaria as text
%         str2double(get(hObject,'String')) returns contents of Aventana_necesaria as a double


% --- Executes during object creation, after setting all properties.
function Aventana_necesaria_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aventana_necesaria (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function Aventana_disponible_Callback(hObject, eventdata, handles)
% hObject    handle to Aventana_disponible (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of Aventana_disponible as text
%         str2double(get(hObject,'String')) returns contents of Aventana_disponible as a double
```

```
% --- Executes during object creation, after setting all properties.
function Aventana_disponible_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aventana_disponible (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Num_capas_Callback(hObject, eventdata, handles)
% hObject    handle to Num_capas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Num_capas as text
%         str2double(get(hObject,'String')) returns contents of Num_capas as a double
```

```
% --- Executes during object creation, after setting all properties.
function Num_capas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Num_capas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Evaluacion_diseno_Callback(hObject, eventdata, handles)
% hObject    handle to Evaluacion_diseno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Evaluacion_diseno as text
%         str2double(get(hObject,'String')) returns contents of Evaluacion_diseno as a double
```

```
% --- Executes during object creation, after setting all properties.
function Evaluacion_diseno_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Evaluacion_diseno (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Boton_CalculoDevanado.
function Boton_CalculoDevanado_Callback(hObject, eventdata, handles)
% hObject    handle to Boton_CalculoDevanado (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Corriente = 3;
Espiras = str2num (get(handles.Espiras,'String'));
R_Eje = str2num (get(handles.R_Eje,'String'));
R_Estator = str2num (get(handles.R_Estator,'String'));
A_Estator = str2num (get(handles.A_Estator,'String'));
LongEstator = str2num (get(handles.L_Estator,'String'));
r_AnilloInterno = str2num (get(handles.r_AnilloInterno,'String'));
R_AnilloInterno = str2num (get(handles.RadAnilloInterno,'String'));
r_AnilloExterno = str2num (get(handles.r_AnilloExterno,'String'));
R_AnilloExterno = str2num (get(handles.RadAnilloExterno,'String'));
R_Disco = str2num (get(handles.R_Disco,'String'));
A_Disco = str2num (get(handles.A_Disco,'String'));

L_Ventana = LongEstator - A_Estator;
Densidad_max = 1;
Factor_empaquetado = 0.9;
S_Espira = Corriente/Densidad_max;
R_Espira = sqrt(S_Espira/pi);
D_Espira = 2*R_Espira;
Espiras_Capa = floor((L_Ventana*Factor_empaquetado)/D_Espira);
Capas = ceil(Espiras/Espiras_Capa);
A_ventana_necesaria = (Capas*D_Espira)/Factor_empaquetado;
A_ventana_disponible = r_AnilloExterno - R_AnilloInterno;
if A_ventana_disponible >= A_ventana_necesaria
    set(handles.Evaluacion_disenio, 'ForegroundColor', 'green');
    set(handles.Evaluacion_disenio, 'String', 'CORRECTO');
else
    set(handles.Evaluacion_disenio, 'ForegroundColor', 'red');
    set(handles.Evaluacion_disenio, 'String', 'INCORRECTO');
end

set(handles.Diametro_espira, 'String', num2str(roundn(D_Espira,-2)));
set(handles.Seccion_espira, 'String', num2str(roundn(S_Espira,-2)));
set(handles.Num_capas, 'String', num2str(roundn(Capas,0)));
set(handles.Aventana_necesaria, 'String', num2str(roundn(A_ventana_necesaria,-2)));
set(handles.Aventana_disponible, 'String', num2str(roundn(A_ventana_disponible,-2)));

```



### 9.3 Código fuente de la función de Matlab “comprobación errores.m” de “FAxial vs Gap.m”

```
function [error] = comprobacion_errores(I, Esp, rEje, Rs, As, Ls, r1, R1, r2, R2, Rd, Ad)
    error = 0;
    if ((Esp < 0) || (rEje < 0) || (Rs < 0) || (As < 0) || (Ls < 0) || (r1 < 0) || (R1 < 0) || (r2 <
0) || (R2 < 0) || ...
        (Rd < 0) || (Ad < 0))
        mostrar_mensaje('Las dimensiones del rodamiento deben tener un valor positivo');
        error = -1;
    else
        if ((Rs < R2) || (Rd < R2))
            mostrar_mensaje('R Estator y R Disco deben ser MAYOR O IGUAL que R Anillo Externo');
            error = -1;
        end
        if ((R2 < r2) || (R2 < R1) || (R2 < r1) || (R2 < rEje))
            mostrar_mensaje('R Anillo Externo debe ser MAYOR que: R Eje, r y R Anillo Interno y r
Anillo Externo');
            error = -1;
        end
        if ((r2 < R1) || (r2 < r1) || (r2 < rEje))
            mostrar_mensaje('r Anillo Externo debe ser MAYOR que: R Eje y r y R Anillo Interno');
            error = -1;
        end
        if ((R1 < r1) || (R1 < rEje))
            mostrar_mensaje('R Anillo Interno debe ser MAYOR que: R Eje y r Anillo Interno');
            error = -1;
        end
        if (r1 < rEje)
            mostrar_mensaje('r Anillo Interno debe ser MAYOR que R Eje');
            error = -1;
        end
        if (Ls < As)
            mostrar_mensaje('L Estator debe ser MAYOR que A Estator');
            error = -1;
        end
    end
end

function mostrar_mensaje(mensaje)
    msgbox(mensaje, 'Error', 'error');
end
```

## 9.4 Código fuente de la función de Matlab “comprobación errores2.m” de “FAxial vs Corriente.m”

```
function [error] = comprobacion_errores2(Esp, rEje, Rs, As, Ls, r1, R1, r2, R2, G, Rd, Ad)
    error = 0;
    if ((Esp < 0) || (rEje < 0) || (Rs < 0) || (As < 0) || (Ls < 0) || (r1 < 0) || (R1 < 0) || (r2 <
0) || (R2 < 0) || ...
        (G < 0) || (Rd < 0) || (Ad < 0))
        mostrar_mensaje('Las dimensiones del rodamiento deben tener un valor positivo');
        error = -1;
    else
        if ((Rs < R2) || (Rd < R2))
            mostrar_mensaje('R Estator y R Disco deben ser MAYOR O IGUAL que R Anillo Externo');
            error = -1;
        end
        if ((R2 < r2) || (R2 < R1) || (R2 < r1) || (R2 < rEje))
            mostrar_mensaje('R Anillo Externo debe ser MAYOR que: R Eje, r y R Anillo Interno y r
Anillo Externo');
            error = -1;
        end
        if ((r2 < R1) || (r2 < r1) || (r2 < rEje))
            mostrar_mensaje('r Anillo Externo debe ser MAYOR que: R Eje y r y R Anillo Interno');
            error = -1;
        end
        if ((R1 < r1) || (R1 < rEje))
            mostrar_mensaje('R Anillo Interno debe ser MAYOR que: R Eje y r Anillo Interno');
            error = -1;
        end
        if (r1 < rEje)
            mostrar_mensaje('r Anillo Interno debe ser MAYOR que R Eje');
            error = -1;
        end
        if (Ls < As)
            mostrar_mensaje('L Estator debe ser MAYOR que A Estator');
            error = -1;
        end
    end
end

function mostrar_mensaje(mensaje)
    msgbox(mensaje, 'Error', 'error');
end
```